HABILITATION À DIRIGER DES RECHERCHES

présentée devant

L'Université de Rennes 1
Institut de Formation Supérieure
en Informatique et Communication

par

Éric Marchand

Contributions à la commande d'une caméra réelle ou virtuelle dans des mondes réels ou virtuels

soutenue le 26 novembre 2004 devant le jury composé de

M.	Patrick Bouthemy	Président
MM.	Ronan Boulic	Rapporteurs
	Radu Horaud	
	Seth Hutchinson	
MM.	Michel Dhome	Examinateurs
	Nassir Navab	
	Kadi Bouatouch	
	François Chaumette	

Table des matières

Remerciements **5**

1 Problématique de recherche **7**

2 Les techniques d'asservissement visuel **13**
2.1 Loi de commande d'asservissement visuel 14
2.2 Tâche hybride : le formalisme de la redondance 17

3 Gestion des mouvements d'une caméra en robotique **21**
3.1 Vision intentionnelle et exploration 22
3.1.1 Vision intentionnelle et active 22
3.1.2 Vision active et exploration d'environnements inconnus 23
3.2 Utilisation de la redondance en asservissement visuel 28
3.2.1 Évitement des butées articulaires 29
3.2.2 Introduction de contraintes exprimées dans l'image 33
3.3 Asservissement visuel robuste aux mesures aberrantes 34

4 Animation d'une caméra virtuelle **43**
4.1 Positionnement et navigation d'une caméra dans des espaces virtuels 44
4.1.1 Spécification des tâches dans l'espace image 45
4.1.2 Introduction de contraintes dans la commande 47
4.2 Application : cinématographie virtuelle 49
4.3 Commande d'un humanoïde de synthèse 52
4.3.1 Techniques d'animation d'un humanoïde 52
4.3.2 Animation à l'aide de l'asservissement visuel 54

5 Suivi d'objets : vers un asservissement visuel virtuel **59**
5.1 Suivi 2D de formes simples pour l'asservissement visuel. 61
5.2 Suivi 3D : localisation d'une caméra. 62
5.3 Suivi hybride 2D-3D 64
5.4 Asservissement visuel virtuel 67
5.4.1 Asservissement visuel virtuel : Principe 68
5.4.2 Asservissement visuel virtuel robuste 69
5.4.3 Suivi d'objets complexes en temps-réel 71
5.5 Autres utilisations possibles pour l'asservissement visuel virtuel 74
5.5.1 Étalonnage de caméra 74
5.5.2 Estimation du déplacement d'une caméra 75
5.6 Application à la réalité augmentée temps-réel 76
5.6.1 Problématique 77
5.6.2 Réalité augmentée à partir de marqueurs 79
5.6.3 Réalité augmentée sans marqueur 80
5.6.4 Réalité augmentée sans modèle 80

6 Bilan et perspectives **85**

A Quelques notations **93**

Bibliographie **94**

Avant propos et remerciements

Les travaux de recherche décrits dans ce mémoire ont été menés entre 1993 et 2004 principalement à l'Irisa/Inria-Rennes. Initialement menés dans le projet Temis (dirigé par Claude Labit) de 1993 à 1996 lors de ma thèse ces recherches se sont poursuivies dans le projet Vista (groupe de Patrick Bouthemy) de 1997 à 2003 et finalement dans l'équipe lagadic (dirigée par François Chaumette) depuis janvier 2004. Par ailleurs, certains des travaux décrits ont été réalisés au département d'informatique de l'Université de Yale (*Artificial Intelligence, Vision & Robotics Laboratory* dirigé par Greg Hager) en 1996 et 1997.

Je voudrais commencer ce document en remerciant ceux avec qui j'ai travaillé directement ou indirectement. J'espère qu'il en auront un souvenir aussi agréable que moi. Par ordre alphabétique, merci à Bruno Arnaldi, Laurent Bonnaud, Nicolas Bonnel, Patrick Bouthemy, Andrew Comport, Nicolas Courty, Grégory Flandin, Greg D. Hager, Nedialko Kostov, Gildas Lefaix, Ezio Malis, Hervé Marchand, Emmanuel Marin, Youcef Mezouar, Valérie Moreau, Michel Perrier, Muriel Pressigout, Yann Ricquebourg, Cédric Riou, Alessandro Rizzo, Jérome Royan, Eric-Paul Rutten. Merci à tous pour les discussions stimulantes et le travail accompli en commun.

Je dois arrêter l'ordre alphabétique et aussi remercier (évidement) François Chaumette et Fabien Spindler. Mes activité de recherche ont été initiées sous la direction de François qui m'a convaincu (sans beaucoup de mal) de l'intérêt de la recherche en vision robotique et est resté par la suite un interlocuteur indispensable. J'ai eu le temps pendant ces (presque) 10 ans d'apprécier ses grandes qualités tant scientifiques qu'humaines. Il est clair que sans Fabien, aucune expérimentation sur les cellules robotiques de l'IRISA ne serait possible et que son travail participe grandement à la validation de tous les travaux présentés dans ce document.

Je souhaiterais finalement remercier les membres du jury qui m'ont fait l'honneur d'évaluer

ces travaux : Seth Hutchinson (professeur à l'Université de l'Illinois à Urbana Champaign), Ronan Boulic (de l'École Polytechnique Fédérale de Lausanne, EPFL) et Radu Horaud (directeur de recherche à l'INRIA Rhones Alpes) à qui j'adresse tous mes remerciements pour le temps passer à analyser, critiquer et commenter ce document en tant que rapporteurs ; Nassir Navab (Professeur à l'université technologique de Münich), Michel Dhome (Directeur de recherche CNRS au Lasmea), et Kadi Boutaouch (professeur à l'université de Rennes 1) pour avoir accepté de participer à ce jury ; et finalement Patrick Bouthemy d'avoir accepté de le présider.

Pour un grand nombre des expérimentations décrites, le lecteur aura la possibilité de juger les résultats non seulement en fonction des explications données dans ce document ou dans les articles donnés en annexes, mais également à partir de documents multimédia accessibles via Internet[1].

J'ai par ailleurs sans doute omis un grand nombre de travaux qui auraient largement mérité d'être cités ici, que leurs auteurs m'en excusent. Concernant mes propres travaux, ce document se voulant synthétique, il ne rend compte que de relativement peu de détails. on pourra cependant trouver dans les articles cités ou fournis en annexe des explications plus complètes. Dans le texte, les références à mes travaux seront notées en italiques (e.g., *[126]*), les autres références apparaîtront quant à elles en caractères normaux (e.g., [173])

[1] http://www.irisa.fr/lagadic/team/marchand/hdr/hdr.html

CHAPITRE 1

Problématique de recherche

Il y a aujourd'hui, d'après l'UNECE[1] environ 770.000 robots opérationnels dont 24.000 en France. À ce jour, la plupart de ces robots sont des manipulateurs industriels dont l'autonomie est souvent réduite au stricte minimum. Ce manque d'autonomie est souvent due à l'absence de capteurs extéroceptifs sur les systèmes robotiques disponibles sur le marché. Ces capteurs sont cependant indispensables des lors que le robot évolue dans un environnement partiellement ou totalement inconnu. Dans ce contexte ces capteurs apportent aux systèmes robotiques les informations nécessaires pour appréhender l'environnement dans lequel ils évoluent. Dans le cadre de nos travaux nous avons uniquement considéré le cas des capteurs de vision qui fournissent au système une information riche mais dégénérée sous la forme d'une projection vers une image 2D de l'ensemble de l'environnement 3D perçu. L'un des points fondamentaux du cycle perception–action est la génération automatique des mouvements d'une ou plusieurs caméras. La création de carte 3D de l'environnement a longtemps été considérée comme un point de passage obligé pour la génération automatique des mouvements d'un robot autonome et de nombreux travaux se sont focalisés, souvent avec succès, sur ce sujet (e.g., [63]). Une façon élégante de raccourcir ce cycle en liant étroitement l'action à la mesure dans l'image est de générer les mouvements du système robotique en ayant recours au techniques d'asservissement visuel.

L'asservissement visuel consiste à contrôler les mouvements d'un système dynamique à partir d'un ensemble d'informations visuelles extraites des images acquises par une ou plusieurs caméras vidéos montées sur (ou observant) l'effecteur d'un robot. Une tâche classique d'asservissement visuel est, par exemple, de réaliser une tâche de positionnement en déplaçant la caméra de façon à ce que l'image perçue corresponde à une image désirée. L'approche développée à l'Irisa et qui est à la base des travaux qui seront décrits dans ce document repose sur la modélisation de fonctions de tâches appropriées et consiste à spécifier le problème en terme de régulation dans l'image d'un ensemble d'informations visuelles.

[1] United Nations Economic Commission for Europe.

Les lois de commande en boucle fermée sur les informations visuelles permettent de compenser les imprécisions des modèles (erreurs de calibration), aussi bien du capteur que du porteur de la caméra. Notre objectif n'a pas été d'étudier en profondeur les aspects théoriques liés à la modélisation de l'information visuelle ou à l'automatique proprement dite mais vise plutôt à l'intégration de l'asservissement visuel au sein de systèmes complexes ou à son utilisation dans des contextes qui ne relèvent pas nécessairement de la robotique et qui ouvrent de nouveaux axes de recherche.

Considérons la tâche de préhension d'un objet dans un environnement encombré et inconnu ou partiellement inconnu. L'asservissement visuel permettra de réaliser la saisie de l'objet à travers la définition d'une liaison rigide entre le couple caméra/effecteur et l'objet à saisir. Un certain nombre d'étapes intermédiaires sont cependant indispensables avant de pouvoir considérer cette saisie. Même en supposant le cas favorable où l'objet est statique, on se rend aisément compte que les problèmes soulevés par cette tâche sont multiples : l'objet est-il visible depuis la position initiale du capteur ? La qualité des images est-elle compatible avec le niveau de performances des algorithmes ? L'information issue du processus de perception est-elle suffisante pour établir la liaison caméra-objet souhaitée ? Si non, quelle tâche de remplacement proposer ? Quelle est la trajectoire du manipulateur permettant d'aller saisir l'objet sans collision avec d'autres objets statiques ou mobiles de l'environnement ? Afin de déployer un tel système en utilisant les techniques d'asservissement visuel, un certain nombre de points devront être considérés :

- Les lois de commande retenues doivent assurer un positionnement correct du système même en cas d'erreurs de modélisation du système (robot, caméra, etc) ou d'erreurs dans l'extraction des indices visuels. Si dans la plupart des cas, des lois de commande relativement simples donnent d'excellents résultats et sont robustes aux erreurs de modélisation, la précision de positionnement est reliée à la précision de l'extraction de l'information visuelle. Réduire la sensibilité au bruit de mesure en travaillant tant au niveau de la commande que du traitement d'image est donc fondamental.
- Par ailleurs, la convergence d'une telle loi de commande n'est en général pas assurée. Les propriétés de convergence et de stabilité découlant du choix de l'information visuelle doivent donc être étudiées avec soin.
- L'asservissement visuel étant une approche locale reposant uniquement sur une perception tronquée et planaire de l'environnement, il est difficile (sauf dans quelques cas particuliers) de prédire ou de contraindre les mouvements 3D du manipulateur. La prise en compte des variations de l'environnement (possibilité d'occultations, d'obstacles) ainsi que des contraintes sur le capteur ou sur le manipulateur est donc un point important.
- Bien évidemment, et même si ce point n'est pas directement relié à la commande proprement dite de la caméra, l'extraction et le suivi des informations visuelles à partir du flux vidéo est un point clé qui conditionnera l'échec ou le succès du déploiement du système.

Notre objectif était donc initialement d'élaborer des stratégies de perception et d'action à partir d'images pour des applications en robotique. Il est cependant apparu que ces techniques ne devait pas se cantonner aux seuls applications robotiques et nous verrons comment elles peuvent être considérées dans des contextes variés comme la vision par ordinateur, la réalité virtuelle et ou encore la réalité augmentée. Ces nouveaux contextes

sont importants en particulier en raison de la variété et du grand nombre des applications potentielles sur lesquelles peuvent ainsi déboucher nos travaux.

Contribution. Une grande partie de nos travaux a porté sur la gestion des mouvements de caméra dans un contexte robotique. Dans la plupart des cas, ces mouvements sont générés par asservissement visuel. Dans cette optique, et même si cette approche apporte en général toute satisfaction, nous nous sommes principalement focalisés sur le développement de techniques permettant d'étendre le champ opérationnel de l'asservissement visuel[2]. Même si les références à la "commande" des mouvements d'une caméra seront nombreuses dans ce document, nos travaux portent cependant principalement sur des aspects de spécification de tâches pour l'asservissement visuel, et non pas sur des aspects d'automatique proprement dite.

L'un des défauts de l'asservissement visuel "de base" est l'absence de contrôle sur la trajectoire du manipulateur. C'est pourquoi afin de pallier l'absence de planification nous avons proposé d'introduire dans la commande des contraintes permettant au système de réagir en cas de modifications locales de l'environnement ou d'éviter des configurations internes indésirables. L'intégration de l'asservissement visuel dans l'approche générale de la fonction de tâche permet de résoudre de manière efficace et élégante les problèmes de redondance rencontrés lorsqu'une tâche visuelle ne contraint pas l'ensemble des degrés de liberté de la caméra. D'une certaine manière, notre objectif a été de descendre les aspects de planification au niveau de la commande, via l'introduction de contraintes compatibles avec la tâche spécifiée en utilisant le formalisme de la redondance. Les travaux que nous avons réalisés sur cette thématique ont porté sur la génération en ligne des mouvements de la caméra afin d'éviter des configurations indésirables du manipulateur par rapport à la scène. Nous avons aussi considéré le problème récurrent en robotique, et qui dépasse le simple contexte de l'asservissement visuel, de l'évitement des singularités et des butées articulaires. Là encore une utilisation originale de la redondance a permis d'apporter une solution élégante.

Un autre frein au déploiement des techniques d'asservissement visuel réside dans le processus d'extraction des informations visuelles. En effet le calcul de la loi de commande est réalisé en minimisant l'erreur entre l'état de primitives visuelles calculé à partir de deux images. Ceci pose d'une part le problème de l'extraction et du suivi de ces primitives visuelles (nous y reviendrons) mais aussi de la sensibilité aux erreurs inhérentes au processus d'extraction des données. L'efficacité de l'asservissement visuel dépend en effet de la précision de l'appariement entre les informations visuelles courante et désirée. Si cette correspondance est entachée d'erreur, la tâche de positionnement sera au mieux imprécise, au pire impossible. Nous avons donc proposé, dans le cadre de la thèse d'Andrew Comport, une modification de la loi de commande permettant de prendre en compte l'incertitude associée aux primitives visuelles et éventuellement de les rejeter.

Nous avons également souhaité revisiter des problèmes classiques en animation par ordi-

[2]Précisons cependant que certains des travaux réalisés ne seront pas du tout décrit dans ce document.

nateur et réalité virtuelle ou en analyse de scènes (localisation, étalonnage, suivi d'objets) à travers l'approche d'asservissement visuel.

Les liens entre la robotique et la réalité virtuelle sont nombreux et anciens (planification des mouvements d'entités virtuelles, animation d'humanoïde reposant sur les techniques de cinématique directe ou inverse, simulation de systèmes dynamiques, assistance à la téléopération). Dans le cadre de la thèse de Nicolas Courty, nous avons débuté une étude portant sur les liens entre l'animation et l'asservissement visuel. Nous avons cherché à proposer à l'animateur une nouvelle modalité d'interface plus "intuitive" avec le monde virtuel en ne spécifiant plus une tâche dans l'espace 3D mais en spécifiant dans l'image produite la position des différents objets, charge au système de prendre en compte en temps réel les différentes contraintes sous-jacentes et de définir la position adéquate de la caméra. En mettant à profit les travaux réalisés dans le domaine de la robotique, l'asservissement visuel est apparu comme une solution efficace pour gérer le problème du déplacement d'entités virtuelles (caméras ou humanoïdes) dans des mondes virtuels et l'animateur peut donc disposer de nouvelles "briques de base" importantes dans son système d'animation. Dans les deux cas, des applications potentielles se situent dans le domaine de la réalité virtuelle, par exemple pour la réalisation de jeux vidéos.

Le suivi d'objets ou de formes dans une séquence d'images est un problème clé, aussi bien en tant que sujet de recherches en vision par ordinateur, que pour valider et, par la suite, pour transférer nos recherches. En effet, les techniques d'asservissement visuel sont à la base des techniques de commande "bas niveau" qu'il faut intégrer dans des systèmes de plus haut niveau pour en assurer une véritable diffusion. Le développement d'algorithmes robustes et rapides de suivi d'objet dans des séquences d'images a donc été un de nos objectifs principaux. L'utilisation de modèles 2D mais surtout 3D semble être une solution intéressante pour parvenir à cet objectif. L'avantage principal des méthodes basées sur un modèle 3D de l'objet est dû au fait que la connaissance *a priori* sur la scène (l'information 3D implicite) permet l'amélioration de la robustesse et de la performance tout en étant capable de fournir l'information supplémentaire nécessaire pour réduire les effets de données aberrantes présentes éventuellement dans le processus de suivi. Dans un premier temps, nous avions proposé un algorithme de suivi hybride 2D/3D reposant à la fois sur une estimation du mouvement dans l'image et sur un calcul de pose.

Par la suite et dans le cadre, entre autres, des thèses d'Andrew Comport et de Muriel Pressigout, nous avons étudié différents problèmes classiques de vision par ordinateur : étalonnage des caméras, localisation 3D, suivi robuste d'objets rigides, suivi d'objets articulés, estimation d'homographies, etc. Ces différent problèmes, qui peuvent se formuler sous la forme d'une erreur à minimiser dans l'image, ont été résolus en utilisant le principe d'une caméra virtuelle, limitée à un simple plan de projection et un centre optique, et dont le déplacement est assuré par des lois de commande d'asservissement visuel. Cet *asservissement visuel virtuel* peut donc être considéré comme le dual de l'asservissement visuel classique. Cette méthode, consistant en fait à minimiser une fonctionnelle non-linéaire en utilisant une approche de type gradient ou quasi-Newton, est donc théoriquement similaire aux méthodes de minimisation classiquement utilisées pour réaliser ce calcul. Le gain par

rapport à ces dernières se situe donc à la fois sur la simplicité de la modélisation et donc de la mise en œuvre, sur la richesse de l'information utilisable, sur son évolutivité (les autres problèmes de vision par ordinateur que nous avons mentionnés peuvent se reformuler de façon similaire), etc.

Il faut bien reconnaître que le développement de ces techniques de localisation et de suivi 3D a été initié par un besoin applicatif. Il est apparu évident au milieu des années 90 que le recours à des objets marqués pour effectuer des tâche de positionnement était un des freins au développement de l'asservissement visuel. L'essor de ces techniques en milieu industriel passe nécessairement par l'extraction et le suivi en temps-réel (50Hz ou mieux si l'on utilise des caméras rapides) d'indices visuels sur lesquels les différentes lois de commande d'asservissement visuel pourront s'appuyer. Nos travaux tentent d'apporter leur contribution à la résolution de ce problème. Des applications de ces algorithmes de localisation et trajectographie 3D existent aussi dans le domaine de la réalité augmentée. Ce champ d'applications est très prometteur, car en plein essor pour la réalisation d'effets spéciaux dans le domaine multimédia ou pour la conception et l'inspection d'objets manufacturés dans le monde industriel.

La robotique, et donc l'asservissement visuel, requiert évidemment une étape de validation expérimentale. Tous les algorithmes présentés dans ce mémoires ont été implantés et testés sur les plate-formes dont nous disposons à l'Irisa. Il nous semble en effet inconcevable dans notre domaine de ne pas valider nos résultats de recherche par des expérimentations sur des systèmes réels. Ces expérimentations sont aussi parfois source d'inspiration pour résoudre de nouveaux problèmes, découverts en observant des propriétés *a priori* impossibles à analyser de manière théorique.

En parallèle des recherches qui seront présentées dans la suite de ce document, nous avons donc également mené des activités importantes de développement de logiciels, soit pour valider nos travaux de recherche via des simulations ou des expérimentations menées sur les plates-formes, soit pour les pérenniser au sein de démonstrations, soit encore dans le cadre de nos activités contractuelles. Nous avons aussi pour objectif de développer un environnement logiciel qui permette le prototypage rapide de tâches d'asservissement visuel. Cet objectif est d'envergure en raison de l'emploi de matériels spécifiques (robot, carte d'acquisition d'images, etc.), mais aussi en raison de la très grande variété des applications potentielles, des lois de commande possibles, et des traitements d'images correspondants. Sans un tel environnement, la mise en œuvre d'applications serait lourde et donc source potentielle d'erreurs. ViSP (pour "Visual servoing platform"), le logiciel prototype développé depuis maintenant six ans, repose sur la mise à la disposition du programmeur d'un ensemble de briques élémentaires qui peuvent être combinées pour construire des applications plus complexes.

Ce mémoire est structuré en quatre chapitres qui portent respectivement sur les thèmes suivants :

- la première partie de ce document présentera un panorama général des techniques d'asservissement visuel ;
- nous décrirons ensuite quelques une nos contributions à ce domaine en nous foca-

lisant principalement sur trois points : la vision intentionnelle pour l'exploration de scène, l'utilisation de la redondance pour introduire des contraintes dans les trajectoires de l'effecteur, et la proposition d'une loi de commande robuste aux données aberrantes ;
- nous motiverons et décrirons ensuite les liens entre l'asservissement visuel et l'animation de scènes dans des environnements virtuels ;
- finalement nous présenterons quelques uns de nos travaux en vision temps-réel.

CHAPITRE 2

Les techniques d'asservissement visuel

L'asservissement visuel [62, 78, 87, 28] consiste à contrôler les mouvements d'un système dynamique en utilisant les informations fournies par une ou plusieurs caméras (ou plus généralement un capteur de vision). Nos travaux reposent sur une approche qui consiste à spécifier une tâche (en général une tâche de positionnement ou de poursuite) comme la régulation *dans l'image* d'un ensemble de caractéristiques visuelles. L'une des difficultés intéressantes de cette approche est que, si l'information utilisée est principalement 2D, les mouvements du système sont générés en 3D. Elle permet en outre de compenser les imprécisions des modèles, aussi bien du capteur que du système à commander, par des lois de commande robustes en boucle fermée sur les informations visuelles extraites de l'image.

On peut ainsi réaliser une grande variété de tâches de positionnement du système par rapport à son environnement en contrôlant entre un et l'ensemble des degrés de liberté du système. Quelle que soit la configuration du capteur, pouvant aller d'une caméra embarquée sur l'effecteur d'un robot à plusieurs caméras déportées, il s'agit de sélectionner au mieux un ensemble d'informations visuelles $\mathbf{s}$ à partir des mesures disponibles dans l'image. Il est ensuite possible d'élaborer une loi de commande contrôlant les degrés de liberté souhaités afin que ces informations $\mathbf{s}$ atteignent une valeur désirée ou consigne $\mathbf{s}^*$ définissant une réalisation correcte de la tâche.

Cette section a pour but de décrire le principe général sur lequel repose l'asservissement visuel. Elle n'a pas pour vocation de fournir un état de l'art exhaustif du domaine (le lecteur intéressé pourra consulter [87] pour un état de l'art jusqu'en 1995, [28] et [29] pour une synthèse des travaux effectués jusqu'en, respectivement, 2000 et 2003, ainsi que les récents numéros spéciaux sur l'asservissement visuel des revues *Int. Journal of Computer Vision* parue en juin 2000 et *Int. Journal of Robotics Research* paru en octobre 2003). Précisons par ailleurs que les notations utilisées dans la suite de ce document sont reprises dans l'annexe A.

2.1 Loi de commande d'asservissement visuel

Les techniques d'asservissement visuel [87, 28] utilisent généralement des informations visuelles de nature 2D, 2D 1/2 ou 3D extraites de l'image. Comme nous l'avons déjà dit, les lois de commande consistent à contrôler le mouvement d'un système dynamique afin que les informations visuelles calculées à partir des mesures dans l'image $\mathbf{s}(\mathbf{r})$, où $\mathbf{r}$ définit la position ou pose de la caméra par rapport à la scène, atteignent une valeur désirée (ou consigne) $\mathbf{s}^*$ ou suivent une trajectoire spécifiée $\mathbf{s}^*(t)$. Ceci revient à minimiser l'erreur :

$$\Delta = \mathbf{s}(\mathbf{r}) - \mathbf{s}^* \tag{2.1}$$

Matrice d'interaction. Afin d'élaborer une loi de commande en boucle fermée sur des mesures $\mathbf{s}$, il est nécessaire d'estimer ou d'approximer la relation qui lie la variation de $\mathbf{s}$ aux variables de contrôle. En dérivant $\mathbf{s}(\mathbf{r})$ par rapport au temps on obtient :

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} = \mathbf{L}_{\mathbf{s}}(\mathbf{s}, Z)\ \mathbf{v} \tag{2.2}$$

où $\mathbf{v}$ est le torseur cinématique de la caméra et où $\mathbf{L}_{\mathbf{s}}(\mathbf{s}, Z)$ est la matrice d'interaction associée à $\mathbf{s}$. Cette matrice dépend de la valeur courante de $\mathbf{s}$, mais aussi de la profondeur de l'objet considéré, représentée par les paramètres notés Z (par la suite et pour simplifier les notations, on notera souvent cette matrice d'interaction $\mathbf{L}_{\mathbf{s}}$).

Si l'on considère une caméra montée sur l'effecteur d'un robot manipulateur, le lien entre $\dot{\mathbf{s}}$ et la vitesse des variables articulaires $\dot{\mathbf{q}}$ du robot s'obtient aisément :

$$\dot{\mathbf{s}} = \mathbf{L}_{\mathbf{s}}(\mathbf{s}, Z)\ \mathbf{J}(\mathbf{q})\ \dot{\mathbf{q}} \tag{2.3}$$

où $\mathbf{J}(\mathbf{q})$ est le Jacobien du robot.

Commande. L'utilisation de l'approche fonction de tâche [173, 62] pour la régulation de cette erreur Δ permet de considérer des résultats généraux pour la synthèse et l'analyse de lois de commande référencées capteurs en boucle fermée. Dans ce contexte, ce problème peut se réécrire sous la forme d'une fonction de tâche à minimiser :

$$\mathbf{e} = \mathbf{C}\big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big) \tag{2.4}$$

où $\mathbf{C}$ est une matrice $6 \times k$ de rang plein (dite matrice de combinaison) qui permet de prendre en compte dans $\mathbf{s}$ un nombre k d'informations visuelles plus important que le nombre maximum de degrés de liberté contrôlables (6 dans notre cas).

De nombreux types de lois de commande ont été proposés dans la littérature : loi de commande non-linéaire [79], optimale de type LQ [80] ou LQP, basée sur des contrôleurs GPC [68] ou H_∞. On peut aussi se limiter à spécifier une décroissance exponentielle découplée de la fonction de tâche :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e}, \tag{2.5}$$

où λ est un gain scalaire qui permet de régler la vitesse de convergence. Il est ensuite assez immédiat d'élaborer une loi de commande générique tentant de réaliser une décroissance exponentielle de l'erreur. En choisissant $\mathbf{C}$ constante, la différentielle de $\mathbf{e}$ est donnée par

$$\dot{\mathbf{e}} = \mathbf{C}\,\dot{\mathbf{s}} = \mathbf{C}\,\mathbf{L_s}\,\mathbf{v}. \tag{2.6}$$

En utilisant (2.5) et (2.6) on obtient une loi de commande idéale donnée par :

$$\mathbf{v} = -\lambda\,\left(\mathbf{C}\,\mathbf{L_s}\right)^{-1}\mathbf{e} \tag{2.7}$$

On a vu que la matrice d'interaction $\mathbf{L_s}$ dépendait non seulement des informations visuelles $\mathbf{s}$ mais aussi de la profondeur relative entre la caméra et l'objet d'intérêt Z. Cette profondeur ne peut être connue parfaitement. Elle ne peut en effet n'être qu'au mieux estimée ou approximée. Dans tous les cas cette connaissance sera entachée d'erreurs et seule une approximation $\widehat{\mathbf{L_s}}$ de $\mathbf{L_s}$ pourra être considérée dans la loi de commande qui en pratique sera donc donnée par :

$$\mathbf{v} = -\lambda\,\left(\mathbf{C}\,\widehat{\mathbf{L_s}}\right)^{-1}\mathbf{C}\big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big) \tag{2.8}$$

Différents choix sont donc possibles pour $\mathbf{C}$ et $\widehat{\mathbf{L_s}}$ en fonction de la taille k de $\mathbf{s}$ mais aussi du comportement désiré du système à contrôler.

Dans le cas le plus simple où la dimension de l'information visuelle est de $k = 6$, on a tout intérêt à choisir $\mathbf{C} = \mathbf{I_6}$. Ce type de loi de commande où l'information visuelle est complète et non redondante est utilisé, par exemple, dans le cas d'asservissement visuel 3D [205, 143] ou encore 2D 1/2 [122, 123][1].

Dans le cas de primitive de nature 2D, le conditionnement de la matrice d'interaction est cependant en général assez mauvais et on préférera utiliser des informations visuelles redondantes (c'est-à-dire où $k > 6$). Dans ce cas, le choix le plus simple est de choisir pour $\mathbf{C}$ la pseudo-inverse d'une approximation de la matrice d'interaction $\mathbf{C} = \widehat{\mathbf{L_s}}^{+}$. On a alors la loi de commande suivante :

$$\mathbf{v} = -\lambda\,\widehat{\mathbf{L_s}}^{+}\big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big) \tag{2.9}$$

Choix du modèle pour la matrice d'interaction. Le choix du modèle sur lequel repose l'approximation $\widehat{\mathbf{L_s}}$ de $\mathbf{L_s}$ est important. Deux choix sont couramment utilisés :

- $\widehat{\mathbf{L_s}} = \mathbf{L}(\mathbf{s}^*, Z^*)$. Dans ce cas, la matrice choisie est constante et correspond à la configuration désirée. Une valeur de profondeur à la position finale, même très approximative, est nécessaire. Ce choix assure la stabilité locale du système parce que la condition de positivité est assurée dans le voisinage de la position désirée. Ceci signifie que, si l'erreur $\mathbf{s}(\mathbf{r}) - \mathbf{s}^*$ est suffisamment petite, la convergence de $\mathbf{s}$ vers $\mathbf{s}^*$ sera obtenue. La matrice d'interaction utilisée est souvent $\widehat{\mathbf{L_s}} = \mathbf{L}(\mathbf{s}^*, \widehat{Z}^*)$, dans ce cas la convergence est aussi sensible aux erreurs potentielles entre Z^* et $\widehat{Z}^*$ [124].

[1] Il ne semble pas exister une sélection de six informations visuelles purement 2D permettant d'assurer une unique solution au problème (il existe généralement 4 situations distinctes entre la caméra et la scène telles que la la projection de ces informations soit la même [58])

– $\widehat{\mathbf{L_s}} = \mathbf{L}(\mathbf{s}, \hat{Z})$. On calcule à chaque itération la valeur courante de la matrice d'interaction. Une estimation des paramètres Z doit alors être réalisée en ligne, par exemple à l'aide de la connaissance d'un modèle 3D de l'objet. Le bruit introduit dans la matrice d'interaction à l'occasion de l'estimation de Z fait que le comportement du système sera généralement moins stable que dans le cas précédent.

Ces deux possibilités ont chacune leurs avantages et leurs inconvénients [27] : dans le premier cas, mouvements de la caméra inadéquats, voire impossibles à réaliser, rencontre éventuelle de minima locaux ; dans le second cas, possible passage de l'objet hors du champ de vue de la caméra. Finalement, il est possible de rencontrer une singularité de la matrice d'interaction, entraînant soit une instabilité de la commande, soit un échec dans la convergence du système.

D'autres choix ont cependant été proposés dans la littérature pour approximer $\mathbf{L_s}$. On trouvera par exemple l'utilisation dans la loi de commande de $\widehat{\mathbf{L_s^T}}$ (au lieu de $\widehat{\mathbf{L_s^+}}$) [80] mais si elle est plus simple d'un point de vue calculatoire, les avantages de cette approche ne sont pas évidents principalement en raison de mauvaises propriétés de découplage. Beaucoup plus récemment il a été proposé d'utiliser la pseudo-inverse de la moyenne entre la matrice $\mathbf{L}(\mathbf{s}^*, Z^*)$ et la matrice $\mathbf{L_s}(\mathbf{s}(\mathbf{r}), \hat{Z})$, ce qui revient à faire une minimisation se rapprochant d'une approche de type Newton mais sans Hessien à calculer ou estimer [121].

Par ailleurs, certains auteurs ont fait le choix d'apprendre la matrice d'interaction. Si l'apprentissage des seuls paramètres inconnus Z ou des paramètres intrinsèques de la caméra peut sembler intéressant [31, 96, 170], l'apprentissage (ou l'estimation numérique) de tous les termes de la matrice d'interaction [182, 85, 89, 184] est critiquable si une expression analytique (même partielle) de la matrice d'interaction est disponible. Si en revanche l'information visuelle est trop complexe (comme les espaces propres [55]) ou dans le cas d'un asservissement visuel déporté ou la position de la caméra par rapport au repère de base du manipulateur serait inconnue [104], cette approche présente un réel intérêt. Par ailleurs, dans certain cas, si de plus cette matrice est judicieusement apprise (voir [110] où ce n'est pas la matrice d'interaction elle même qui est apprise mais son inverse), le comportement du système peut s'avérer plus adéquat et le cône de convergence peut être plus important que dans le cas d'une matrice d'interaction analytique.

Nature de l'information visuelle. Une classification des différents types d'asservissement visuel peut être établie en fonction du type d'informations $\mathbf{s}$ utilisées pour établir la loi de commande :

– **asservissement 3D** (ou *position-based visual servoing*) utilise en entrée de la boucle de commande des informations tridimensionnelles exprimées dans un repère euclidien. Plus précisément, dans le cadre d'une tâche de positionnement rigide (c'est-à-dire utilisant les six degrés de liberté de la caméra), la consigne peut s'exprimer sous la forme d'un déplacement 3D à réaliser [205] ou directement en utilisant les coordonnées de points 3D [143]. Ces informations peuvent être acquises à l'aide de méthodes de calcul de pose (e.g., [56, 117]*[36]* pour les expériences réalisées à l'IRISA *[136]*).
– **asservissement 2D** (ou *image-based visual servoing*). Dans ce cas, sans doute le plus classique, les informations utilisées sont exclusivement extraites de l'image. La consigne est alors exprimée comme la différence entre un motif courant et un

motif désiré dans l'espace image. Ces primitives peuvent être des points, des droites, des cercles, des moments [30], ou tout autre type ou combinaison d'informations géométriques extraites de l'image.

– **asservissement 2D 1/2** [122]. Cette approche utilise comme information à la fois des informations directement exprimées dans l'image et des informations exprimées dans le repère de la caméra. Cette approche permet un fort découplage de la loi de commande et un contrôle partiel dans l'image qui permet de conserver l'objet dans le champ de vue de la caméra. Une étude théorique de la stabilité et du domaine de convergence est par ailleurs possible.
– **asservissement d2D/dt** [35, 174, 51]. Dans ce cas très original, les informations utilisées ne sont plus de nature géométrique (coordonnées de points, déplacement 3D, etc.) mais de nature dynamique. La consigne s'exprime alors comme la régulation du mouvement 2D apparent à un champ de vitesse désiré. Ce type d'asservissement pallie les problèmes liés à l'extraction des primitives visuelles dans l'image.

L'obtention de la consigne $\mathbf{s}^*$ est un point clé dans la mise en œuvre d'une tâche d'asservissement visuel. Dans la plupart des cas, cette consigne est calculée (si l'on dispose d'information *a priori* sur la scène) ou apprise à partir d'une image acquise à la position désirée lors d'une phase d'apprentissage. Dans certains cas, par exemple celui d'un asservissement visuel déporté depuis une caméra mobile, le calcul de cette consigne peut se révéler relativement complexe *[136]*.

La figure 2.1 illustre une expérience classique de positionnement par asservissement visuel 2D (l'algorithme permettant le traitement de telles images sera présenté dans la section 5.4). La position désirée (ou consigne $\mathbf{s}^*$) qui apparaît ici en bleu est préalablement apprise.

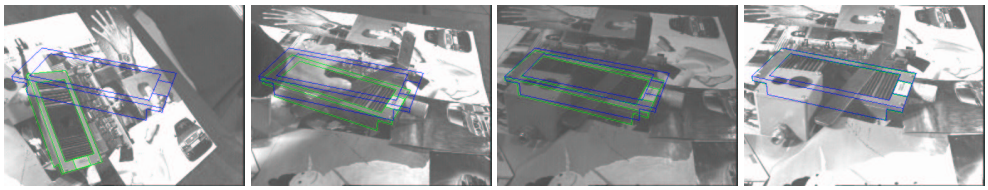


FIG. *2.1: Suivi d'un objet pendant une expérience d'asservissement visuel 2D. L'objet suivi est en vert et sa position désirée dans l'image est en bleu.*

2.2 Tâche hybride : le formalisme de la redondance

La régulation de la tâche visuelle n'est pas toujours l'unique objectif visé et il peut être souhaitable de combiner cette tâche avec une autre telle que, par exemple des opérations de suivi de trajectoires pour des applications d'inspection, ou d'évitement des butées et singularités du robot, d'évitement d'obstacles, etc.

Il est évident que si les 6 degrés de liberté de la caméra sont contraints par la tâche visuelle, cette combinaison est *a priori* impossible et ne peut résulter au mieux que d'un

compromis. Si par contre les 6 degrés de liberté de la caméra ne sont pas contraints par la tâche, il est alors possible de combiner ces deux objectifs. Différentes stratégies sont possibles pour combiner deux tâches. La plus simple consiste à faire une combinaison linéaire des deux tâches. En asservissement visuel, cette stratégie a été largement utilisée principalement par Nelson [155, 154, 156] mais aussi dans [24] pour des tâches d'évitement d'obstacles. Cette stratégie n'est cependant pas optimale puisque l'exécution de la tâche secondaire entraîne une perturbation de la tâche principale. Il est par ailleurs possible que la loi de commande réalise un compromis qui peut aboutir à ce qu'aucune des deux tâches ne soient réalisées. Rien n'assure en effet *a priori* que les deux tâches soient compatibles.

L'intégration de l'asservissement visuel dans l'approche générale de la fonction de tâche [173] permet de résoudre de manière efficace et élégante ce problème de tâche hybride. Dans ce contexte, la tâche visuelle (que l'on notera $\mathbf{e_1}$) est considérée comme principale et prioritaire. Les autres objectifs s'expriment sous la forme d'une tâche secondaire $\mathbf{e_2}$. Cette tâche secondaire peut être définie, par exemple, comme le gradient d'une fonction h_s à minimiser sous la contrainte que la tâche principale soit réalisée. Ceci est possible car, si la pseudo-inverse fournit une commande de norme minimale permettant de réguler l'erreur, il existe en fait une infinité de solutions permettant d'assurer cette minimisation (voir figure 2.2-a). Toutes les autres solutions se situent dans le noyau du Jacobien de la tâche. Si la tâche secondaire est projetée sur ce noyau (voir figure 2.2-b) elle n'aura alors aucun effet sur la tâche principale [114, 173, 62].

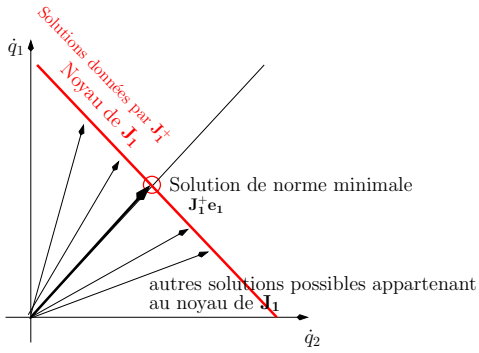


Fig 2.2-a : Il existe a priori *une infinité de solutions aux problèmes, toutes situées sur le noyau $\mathbf{J_1^\perp}$ de $\mathbf{J_1}$. La pseudo-inverse $\mathbf{J_1^+}$ de $\mathbf{J_1}$ fournit la solution de norme minimale.*

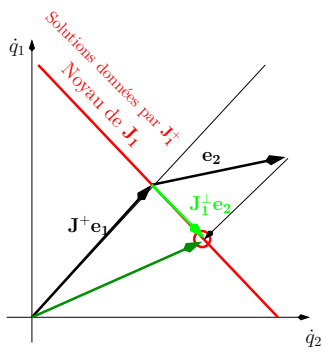


Fig 2.2-b : Dans le cas où une tâche secondaire $\mathbf{e_2}$ est considérée, elle est projetée sur le noyau $\mathbf{J_1^\perp}$ de la tâche et n'a donc aucun effet sur la tâche principale.

FIG. *2.2: Illustration [11] du principe de redondance (pour deux axes q_1 et q_2)*

La tâche $\mathbf{e_1}$ de dimension $m \leq 6$ s'écrit toujours :

$$\mathbf{e_1} = \mathbf{C}\big(\mathbf{s} - \mathbf{s}^*\big)$$

où $\mathbf{C}$ est toujours de rang plein et de dimension $m \times k$. Une fonction de tâche générale qui réalise $\mathbf{e_2}$ sous la contrainte $\mathbf{e_1} = 0$ s'écrit alors [114, 173, 62] :

$$\mathbf{e} = \mathbf{J_1^+e_1} + \alpha\mathbf{J_1^\perp e_2} \tag{2.10}$$

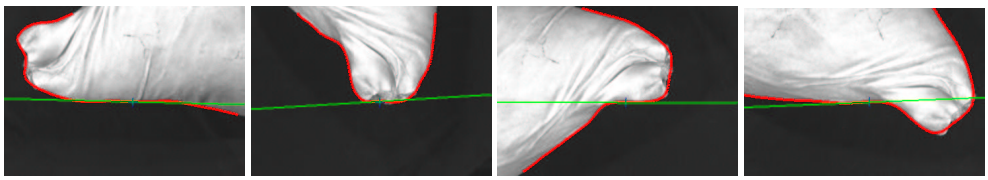


FIG. *2.3: Utilisation de la redondance pour une tâche de suivi de contour. La tâche principale impose que la tangente à la courbe extraite de l'image soit horizontale dans l'image (ce qui contraint 2 degrés de liberté), la tâche secondaire impose un mouvement dans la direction de l'axe des x du repère de la caméra.*

où $\mathbf{J_1} = \mathbf{CL_s}$ est le Jacobien de la tâche $\mathbf{e_1}$ de dimension $m \times n$. $\mathbf{J_1^+}$ et $\mathbf{J_1^\perp} = \mathbf{I_n} - \mathbf{J_1^+J_1}$ sont deux opérateurs de projection orthogonaux qui garantissent que les mouvements de la caméra dus à la tâche secondaire sont compatibles avec la tache principale[2]. α est un scalaire qui définit l'amplitude des mouvements du manipulateur dus à la tâche secondaire (le réglage de ce gain est souvent un problème non trivial). La matrice de combinaison $\mathbf{C}$ doit être choisie de façon à ce que le Jacobien $\mathbf{J_1}$ soit de rang plein.

Une fonction de tâche $\mathbf{e}$ réalisant la minimisation de h_s sous la contrainte $\mathbf{e_1} = 0$ s'écrit sous la forme :

$$\mathbf{e} = \mathbf{W^+e_1} + \alpha(\mathbf{I_n} - \mathbf{W^+W})\mathbf{e_2} \tag{2.11}$$

$\mathbf{W}$ est définie comme une matrice de rang plein telle que Ker $\mathbf{W}$ = Ker $\mathbf{L_s}$ [114, 173, 62]. En raison du choix de $\mathbf{W}$, $\mathbf{I} - \mathbf{W^+W}$ appartient théoriquement au noyau Ker $\widehat{\mathbf{L_s}}$, ce qui implique que la réalisation de la tâche secondaire n'aura aucun effet sur la tâche primaire ($\mathbf{L_s}(\mathbf{I_n} - \mathbf{W^+W})\mathbf{e_2} = \mathbf{0}, \forall \mathbf{e_2}$). Évidemment, si la tâche d'asservissement visuel contraint les 6 degrés de liberté du robot, on a alors $\mathbf{W} = \mathbf{I}$, ce qui implique $\mathbf{I} - \mathbf{W^+W} = \mathbf{0}$. Il est alors impossible de considérer une quelconque tâche secondaire.

D'autre part, en pratique $\mathbf{W}$ ne peut être construit qu'à partir d'une estimation $\widehat{\mathbf{L_s}}$ de la matrice d'interaction (puisque les paramètres Z de la matrice d'interaction sont *a priori* inconnus). L'opérateur de projection $\mathbf{I} - \mathbf{W^+W}$ n'appartiendra donc pas exactement à Ker $\mathbf{L_s}$ et la tâche secondaire introduira une perturbation dans la réalisation de la tâche visuelle.

La loi de commande complète est maintenant donnée par [62, 28] :

$$\mathbf{v} = -\lambda\mathbf{e} - (\mathbf{I_n} - \mathbf{W^+W})\frac{\partial \mathbf{e_2}}{\partial \mathbf{t}} \tag{2.12}$$

Si cette stratégie permettant la combinaison de deux tâches est relativement classique pour la commandes des systèmes dynamiques en robotique [173, 61, 1, 10] ou en animation [11], elle n'a été que rarement utilisée en asservissement visuel autrement que pour pour le contrôle de la vergence d'une tête stéréoscopique [50] ou pour du suivi de trajectoires [62,

[2]Notons que si l'on doit considérer plusieurs tâches secondaires, il est possible, soit de faire une combinaison linéaire de ces plusieurs tâches (mais la remarque concernant la combinaison par une telle approche de la tâche visuelle et de la tâche secondaire s'applique aussi dans ce cas), soit de les hiérarchiser et de projeter chaque tâche sur le noyau de la précédente [10].

31, 42]*[126]*. La figure 2.3 montre une utilisation de la redondance pour réaliser un suivi de contour (ici un jambon). Dans les sections 3 et 4, nous présenterons d'autres utilisations possibles de ce formalisme de la redondance.

CHAPITRE 3

Gestion des mouvements d'une caméra en robotique

De nombreux travaux menés en robotique se sont fixés pour objectif la réalisation de robots autonomes devant évoluer dans des environnements partiellement ou totalement inconnus, devant interagir avec des partenaires humains et devant exécuter un ensemble de tâches plus ou moins complexes (se localiser, se déplacer, saisir et manipuler des objets, etc). De tels systèmes n'existent actuellement, et pour sans doute encore un certain temps, que dans les romans [6] ou les films de science fiction. Développer un tel robot requiert la définition d'un contrôleur de haut niveau devant appréhender la tâche à exécuter dans sa globalité et définissant les stratégies de perception et d'action à mettre en jeu pour parvenir au but. Par ailleurs ce contrôleur doit gérer l'enchaînement d'un ensemble de tâches élémentaires pouvant le plus souvent se réduire à des cycle perception-action (comme par exemple les lois de commande d'asservissement visuel ou plus généralement les lois de commande référencée capteurs).

Toutes les tentatives passées visant à réaliser des solutions universelles aux problèmes de perception se sont soldées par des échecs. Partant de ce constat, certains auteurs ont décrit le paradigme de la vision active visant à définir des systèmes sans doute moins généraux mais plus efficaces. Historiquement, nos premier travaux ont porté sur cette problématique dans un contexte de reconstruction et d'exploration de scène 3D. Dans cette optique, nous avons défini des stratégies de déplacement de caméra reposant d'une part sur des techniques de calcul de points de vue et d'autre part sur la définition de tâches simples d'asservissement visuel. Un contrôleur de haut niveau assurait l'enchaînement adéquat de tous ces processus élémentaires afin d'assurer l'exécution de la tâche nominale : la reconstruction complète de la scène.

En développant ce système il est vite apparu que si la définition de ces stratégies de haut niveau était fondamentale, le système était en pratique trop souvent mis en échec par une mauvaise exécution ou un échec des tâches élémentaires. Fondamentalement, les causes principales de l'échec du processus provenaient d'une part de l'absence de planification

de la trajectoire du manipulateur pendant les phases de positionnement et d'autre part des difficultés issues du processus de traitement d'image. Nous avons donc tenté d'apporter un début de solution à ces deux points dans la suite de nos travaux. Les solutions visant à l'introduction de contraintes, portant sur le système ou sur l'environnement, dans la trajectoire de la caméra pendant une phase d'asservissement visuel, reposent sur l'utilisation de la redondance. Les solutions aux problèmes de l'extraction de l'information visuelle reposent quant à elles sur deux stratégies différentes mais complémentaires : la première, évidente, repose sur le développement d'algorithmes de suivi d'objets fiables (et sera décrite dans le chapitre 5), la seconde repose sur une modification de la loi de commande décrite dans la section précédente afin de la rendre robuste à la présence de données aberrantes.

3.1 Vision intentionnelle et exploration

3.1.1 Vision intentionnelle et active

Dans une optique d'analyse de scènes, les approches de vision inspirées du paradigme de Marr [141] considèrent un capteur généralement statique, éventuellement mobile, mais non contrôlé. Cette approche s'avère insuffisante pour résoudre un grand nombre de problèmes où une modification pertinente des paramètres intrinsèques et/ou extrinsèques du capteur est nécessaire. C'est pourquoi Aloimonos [3], [2], Bajcsy [12] ou encore Ballard [13] ont proposé de modifier radicalement cet état de fait en élaborant le concept de vision active [185]. Les techniques de vision active tirent leur origine d'une tentative de simulation du système visuel animal et humain en essayant de recréer ses facultés d'adaptation. D'un point de vue méthodologique, la vision active, où les informations perçues sont utilisées au sein d'une boucle de rétroaction, tente surtout d'améliorer la qualité de la perception par rapport à l'approche passive classique, où l'on se restreint à observer, mesurer et interpréter les données issues du capteur. La vision active consiste en effet à élaborer des stratégies de perception intelligentes, en contrôlant les paramètres du capteur (position, vitesse, mise au point, etc.). Elle peut être définie comme un processus d'acquisition "intelligent" des données afin de résoudre les problèmes soulevés lors de la conception d'un système de vision par ordinateur, à savoir leur sensibilité au bruit, leur faible précision et surtout leur manque de réactivité.

Par principe, les travaux réalisés dans le domaine de la vision active sont moins ambitieux, puisque dédiés à une application ou un but précis (la vision est alors intentionnelle) dans un cadre déterminé. Il semble évident que les stratégies à élaborer sont différentes d'une application à l'autre. Ainsi, si l'on cherche à reconstruire un objet pour une application de préhension ou d'évitement d'obstacles, la différence de précision des résultats souhaités entraînera l'utilisation de méthodes plus ou moins élaborées et contraignant plus ou moins les mouvements du capteur, pour atteindre, et atteindre seulement, la précision demandée. Les stratégies dépendent également très fortement des ressources disponibles. Par exemple, si l'on utilise un robot manipulateur, ou au contraire un robot mobile, comme moyen de déplacement du capteur, les stratégies pour résoudre le même problème seront généralement très différentes en raison des problèmes d'odométrie que l'on rencontre en robotique mobile et qui sont quasiment inexistants pour la plupart des robots manipulateurs.

L'inconvénient majeur de la vision active est donc l'absence de généricité des tra-

vaux qui découlent de ce concept pourtant générique. Cependant, des classes de méthodes, comme par exemple les techniques d'asservissement visuel ou de prédiction et vérification d'hypothèses, semblent particulièrement bien adaptées à la vision active. C'est pourquoi nous avons tenté de suivre une méthodologie qui devrait permettre de considérer une vaste classe de problèmes liés à la réalisation de tâches robotiques en environnement statique ou dynamique à l'aide d'informations visuelles. Cette méthodologie repose sur les trois relations suivantes :

- *la relation entre le global et le local.* Une tâche est généralement définie de manière globale (par son but). Il s'avère cependant que les informations disponibles pour parvenir à ce but sont généralement locales. La relation entre cette modélisation globale du but et cet ensemble de sous-modèles locaux, dépendant fortement de la localisation et des paramètres de la caméra, doit donc être étudiée afin de réaliser la tâche spécifiée.
- *la relation entre le continu et le discret.* Cet aspect du problème est très fortement relié au précédent. Si la mise en œuvre des tâches élémentaires s'appuie le plus souvent sur des méthodes continues, comme les lois de commandes qui contrôlent les mouvements du capteur, l'enchaînement des différentes tâches menant à la réalisation de la tâche nominale repose sur la manipulation d'informations logiques, temporelles, etc. Les mouvements de la caméra peuvent être gérés de façon continue en utilisant des techniques issues de l'automatique tel que l'asservissement visuel, ou discrète en utilisant, par exemple, les techniques de calcul de points de vue.
- *la relation entre la perception et l'action.* Le point fondamental de l'approche proposée est la relation existant entre le mouvement du capteur et les informations perçues pendant ce mouvement. L'information permet de guider le capteur dans son déplacement lorsque le déplacement sert à acquérir l'information. Cette boucle de rétroaction, qui peut paraître naturelle, se retrouve cependant assez rarement abordée dans la littérature. Elle nous semble pourtant fondamentale dans un système de vision active. Cette boucle se retrouve à tous les niveaux dans les différents systèmes que nous avons proposés (i.e., tant au niveau local que global, continu que discret).

3.1.2 Vision active et exploration d'environnements inconnus

De nombreux travaux menés en vision artificielle se sont fixés pour objectif la réalisation de systèmes puissants capables d'accéder à la géométrie spatiale d'une scène à partir de son observation par une caméra mobile. Ces systèmes doivent fournir une description 3D géométrique claire et complète de la scène à partir d'une séquence d'images 2D. Dans *[132, 131]*, nous avons traité le problème de la reconstruction d'environnements assez restreints (objets statiques, informations *a priori* sur la nature des objets constituant la scène,...). L'approche que nous avons retenue dans ces travaux pour la reconstruction consiste à estimer les paramètres décrivant la structure spatiale d'une primitive géométrique 3D par des techniques de "structure à partir du mouvement" [31]. Cette technique repose sur une analyse du déplacement apparent de la primitive dans la séquence d'images et sur la mesure des mouvements de la caméra. Afin de prendre en compte les erreurs de mesure qui perturbent le processus de reconstruction ainsi que le biais dans l'estimation dû aux erreurs de discrétisation inhérentes à ce type d'approche, une méthode d'optimisation

par vision active est décrite dans [17, 31], les mouvements de la caméra étant automatiquement générés par asservissement visuel. Les résultats obtenus par cette approche active sont beaucoup plus fiables, robustes et précis que ceux obtenus sans contrôle explicite du mouvement de la caméra (vision dynamique).

Cette approche ne permet cependant de reconstruire qu'une seule primitive à la fois ; de plus, une connaissance *a priori* sur la nature de la primitive est nécessaire afin de générer les mouvements optimaux de la caméra. L'objectif a donc été de s'abstraire de ces contraintes par la définition de stratégies de perception de la scène afin d'aboutir à une représentation 3D précise et complète de la zone à reconstruire. De manière schématique, l'approche utilisée consiste à découvrir et sélectionner automatiquement les informations pertinentes, puis par des phases d'exploration nécessaires à la complétude de la reconstruction, à se focaliser successivement sur les différents objets de la scène.

À cet aspect *local et continu* du processus de reconstruction que constitue l'estimation des paramètres des primitives, il est nécessaire de superposer une reconstruction incrémentale qui correspond aux stratégies de reconstruction et d'exploration de la scène 3D. Cette reconstruction est de caractère *événementiel* et est pilotée par la découverte de nouvelles primitives dans l'image. L'approche que nous avons définie pour la reconstruction de scènes complexes consiste à sélectionner *automatiquement* les informations images pertinentes puis à focaliser successivement la caméra sur les différentes primitives de la scène afin de les reconnaître et ensuite de les reconstruire. Outre ces phases de focalisation et de reconstruction des primitives observées, des phases d'exploration sont introduites afin de considérer l'ensemble des primitives de la scène même si celles-ci ne sont pas initialement visibles. Schématiquement, la reconstruction de la scène se fait donc en deux étapes principales :

- La première étape, qui inclut la reconstruction 3D, permet de reconstruire de manière incrémentale l'ensemble des primitives qui apparaissent dans le champ de vision de la caméra. Cette phase est dites d'**exploration locale** car elle ne fait appel qu'à des informations disponibles localement.
- Quand toutes les primitives initialement observées ont été reconstruites, une stratégie différente est mise en œuvre afin de focaliser la caméra sur des zones de la scène n'ayant pas encore été observées. Il s'agit alors d'**exploration globale**. Cette étape a pour objectif d'assurer une reconstruction aussi complète que possible de la scène.

Exploration locale - Réseaux Bayesiens pour la prédiction/vérification. Nous avons développé un algorithme, simple et efficace, qui permet de reconstruire de façon incrémentale toutes les primitives qui ont été observées par la caméra *[132]*. Cet algorithme repose sur le fait que la projection dans l'image des primitives qui nous intéressent sont des segments. Tous les segments observés dans l'image font l'objet d'une phase de reconstruction (à savoir reconnaissance et reconstruction de la primitive 3D associée). L'algorithme permet d'assurer que chaque primitive sera reconstruite une fois et une seule. On obtient ainsi une modélisation 3D d'une partie de la scène considérée comprenant également les zones libres (calculées par lancer de rayons). Cette modélisation reste cependant une modélisation de bas niveau et est parfois incomplète. Concernant les segments, nous souhaitons passer à une représentation hiérarchique en terme de segments 3D, jonctions 3D, polygones et dans la mesure du possible faces, etc.

La méthode développée dans ce but vient se greffer sur l'algorithme de reconstruction incrémentale et repose sur des techniques de **prédiction / vérification d'hypothèses**. Du fait des incertitudes dans les mesures et dans les observations, nous avons utilisé une approche probabiliste. La génération d'une hypothèse se fait en utilisant bien sûr le modèle 3D courant, mais aussi en se basant sur un certain nombre d'*a priori* très généraux sur les caractéristiques d'une scène polyédrique. Les connaissances que nous avons introduites sont codées dans des réseaux Bayesiens. Les réseaux Bayesiens se prêtent en effet très bien au raisonnement et à la prise de décision en présence d'incertitude [162]. Dans notre cas, ces réseaux permettent d'émettre des hypothèses sur l'existence et la localisation de nouveaux objets, puis de proposer l'exécution d'une action conduisant à vérifier ou à infirmer cette hypothèse, enfin, en fonction du résultat de l'étape de vérification, de compléter le modèle 3D de la scène. L'étape de vérification s'appuie à la fois sur les observations déjà réalisées sur la scène, mais aussi sur une acquisition d'informations nouvelles nécessitant un déplacement du capteur (déplacement qui est alors automatiquement réalisé par asservissement visuel, voir par exemple sur la figure 3.1). Enfin, la modélisation de la scène et la création de nouveaux objets (jonctions, polygones, ...) reposent sur les informations 3D et 2D provenant de l'ensemble du processus de reconstruction déjà réalisé et sur l'apport d'informations introduites par les hypothèses validées.

L'utilisation de cette approche nous permet de disposer d'une modélisation de la scène en terme d'objets et non plus en terme de primitives simples comme les segments 3D. Cette modélisation beaucoup plus riche sert de base à l'exploration globale de la scène (voir un exemple de reconstruction sur la figure 3.2).

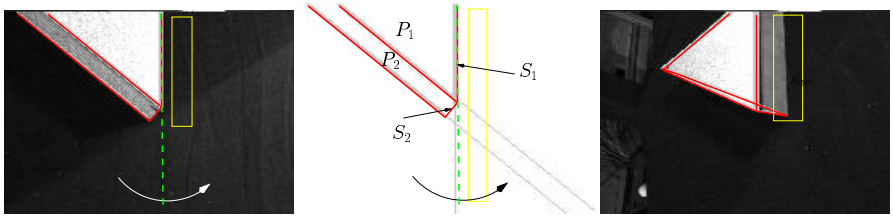


FIG. *3.1: Exemple d'un processus de prédiction/vérification. L'utilisation de techniques Bayesiennes a permis de prédire la position d'un segment (en fonction du reste des connaissances 3D disponibles). Un mouvement de la caméra est généré (par asservissement visuel) pour vérifier cette hypothèse.*

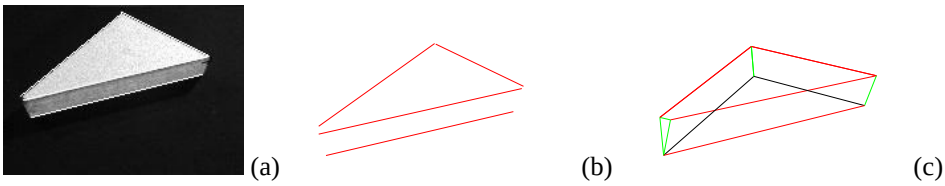


FIG. *3.2: Autre exemple d'un processus de prédiction/vérification. (a) Image initiale de la scène, (b) modèle 3D acquis en se servant uniquement du module de reconstruction incrémentale, (c) modèle reconstruit en se servant du module de prédiction/vérification d'hypothèses*

Exploration globale - Complétude de la reconstruction. Si, après une phase d'exploration *locale*, il est possible d'assurer que toutes les primitives observées par la caméra ont été reconstruites, il est par contre impossible d'affirmer, à ce stade, que tous les objets composant la scène ont été traités. Compléter le modèle géométrique requiert des mouvements exploratoires permettant d'observer les parties cachées et/ou les parties non encore observées de la scène. Cette exploration passe par le calcul d'un certain nombre de points de vue qui permettront l'observation de nouveaux objets ou au contraire qui permettront d'assurer que telle ou telle partie de l'espace est vide. La méthode que nous avons développée s'inspire des travaux décrits dans [40, 207, 195]. On peut également signaler les travaux sur l'exploration complète d'un seul objet [107], la découverte des zones occultées [145], la recherche d'un objet donné dans un environnement [166, 208], l'exploration autonome d'une scène basée sur la fusion de données imprécises [151, 204], ainsi que les travaux sur l'observation optimale d'un objet (qui reposent sur une connaissance a priori de la scène) [48, 188, 187].

Le problème du calcul de points de vue est un problème difficile et assez peu étudié. Généralement, les solutions proposées reposent sur une connaissance *a priori* de la scène. Dans notre cas, les connaissances sur la scène sont presque nulles. La stratégie de calcul de points de vue que nous avons mise en œuvre repose sur la modélisation et l'optimisation d'une fonction de coût codant au mieux la tâche que nous souhaitons effectuer. Nous avons retenu quatre critères associés au point de vue, qui sont intégrés dans cette fonction. En premier lieu, nous nous basons sur le gain apporté par une nouvelle position. Ce gain détermine le volume potentiel découvert calculé en utilisant les techniques de lancer de rayons. Un critère modélisant le coût du déplacement d'un point de vue au suivant est aussi justifié par le fait que nous souhaitons minimiser la distance totale parcourue par la caméra. Les contraintes mécaniques du robot conduisent également à l'introduction d'un critère éloignant le robot de ses butées articulaires. Enfin, les connaissances déjà acquises sur la scène permettent de définir un critère binaire représentant l'accessibilité d'un point de vue. La minimisation d'une fonction intégrant ces différents critères permet le calcul d'un nouveau point de vue *[131]*. Cette minimisation est effectuée à l'aide d'un ICM multi-échelle préférable à l'emploi d'une méthode stochastique de type recuit simulé, notamment pour des raisons de temps de calcul. Initialement, la caméra se déplace sur une demi-sphère englobant la scène pour éviter les obstacles potentiels (encore inconnus). Dès que des zones accessibles sont reconstruites, la caméra peut se déplacer à l'intérieur de ces zones situées dans la demi-sphère. Un algorithme de focalisation sur les zones inconnues résiduelles a également été proposé *[129]* ; il permet de traiter de manière adéquate les parties occultées par des objets (voir un exemple de reconstruction sur la figure 3.3). Ajoutons finalement que nous avons proposé des techniques issues de la programmation dynamique qui permettent de minimiser (sous certaines hypothèses) le nombre de points de vue *[129]*. L'exploration de la scène s'achève quand, quel que soit le point de vue choisi parmi tous les points de vue accessibles, il n'y aura plus d'apport supplémentaire d'informations. Ceci signifie que la reconstruction sera alors aussi complète que possible compte tenu des contraintes imposées par le manipulateur et/ou par la scène. Cette méthode d'exploration de scènes a été développée et testée dans le cadre du processus de reconstruction par vision active présenté dans [31]. Cependant, elle est indépendante de la méthode de reconstruction choisie et est applicable dès lors que l'on peut disposer d'une représentation dense des zones observées

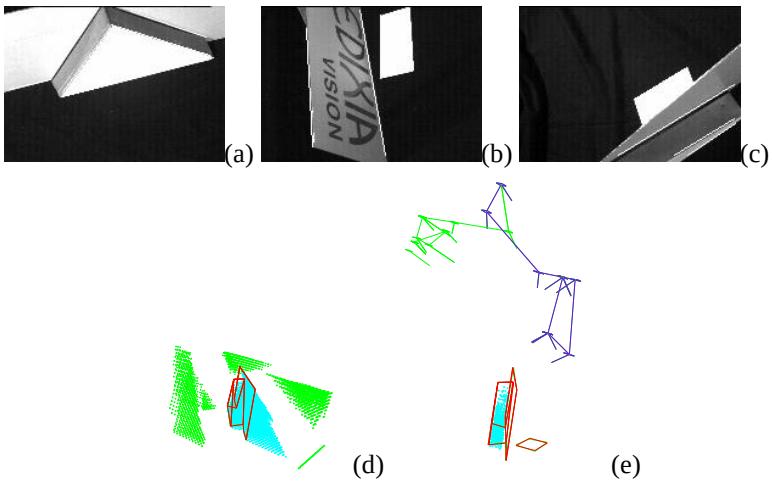


FIG. *3.3: Résultat d'un processus d'exploration globale par calcul de point de vue : (a) première image acquise par le système, (b) image de la même scène acquise pendant l'exploration (de nouvelle primitives apparaissent), (c) reconstruction et zones inconnues après la première reconstruction incrémentale, (d) zones connues et inconnues avant le début du processus d'exploration, (e) différent points de vue calculés pendant l'exploration (il ne reste plus de zones inconnues)*

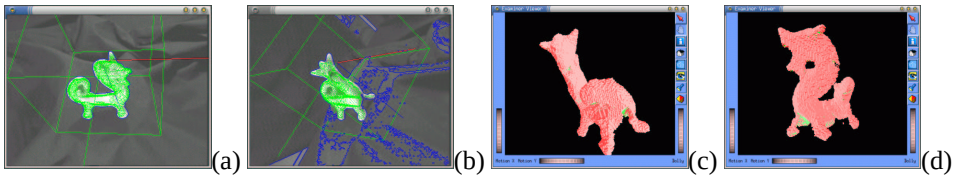


FIG. *3.4: Reconstruction d'un objet 3D par des techniques de coloration de voxels et exploration par asservissement visuel. (a-b) images acquises depuis positions différente de la caméra avec projection du modèle reconstruit (c-d) deux vues du modèle final.*

(par stéréovision dense, "coloration de voxels"[175], *space carving* [108], construction de l'enveloppe visuelle [144, 21], capteur laser, etc.).

Par la suite, et dans le cadre d'un contrat pour l'Ofival portant sur l'évaluation de la qualité des pièces de viandes de porc par vision active et mesures IRM, nous avons en collaboration avec les CEMAGREF développé un système de reconstruction 3D (reposant sur des techniques de "coloration de voxels") utilisant une caméra commandée automatiquement par asservissement visuel pour centrer l'objet à reconstruire tout en se déplaçant afin de maximiser la quantité d'information nouvelle acquise [127] (voir figure 3.4).

Si la notion de vision active ou intentionnelle était très en vogue à la fin des années 80 et dans le courant des années 90, ce n'est semble-t-il plus réellement le cas en ce début de XXIème siècle (pour le moins, sous cette appellation). Il faut bien reconnaître que malgré quelques beaux résultats théoriques, les méthodes décrites dans la littérature manquent dans la plupart des cas de généricité. Il n'en demeure pas moins que les briques algorithmiques élémentaires de ces systèmes (calcul de point de vue, réseaux Bayesiens, asservissement

visuel dans notre cas) restent des éléments actuellement couramment utilisés pour l'intégration de systèmes de vision robotique.

Par ailleurs, même si les mouvements de la caméra étaient, à un niveau local, générés par asservissement visuel, nous avions principalement considéré dans le cadre de ce travail un processus de déplacement discret de la caméra (calcul de nouveaux points de vue). Pour définir ces points de vue, un certain nombre de contraintes était introduites dans une fonction de coût qu'il convenait ensuite de minimiser. Il est cependant légitime de se demander si de telles stratégies de calcul de points de vue trouvent encore leur place dans l'optique de réalisation de tâches robotiques diverses dans un environnement potentiellement dynamique perçu par une ou plusieurs caméras (c-à-d, objets mobiles dont la trajectoire est *a priori* inconnue). Dans la suite de nos travaux nous avons donc étudié l'introduction de telles contraintes directement dans les lois de commande d'asservissement visuel (c'est-à-dire plus à un niveau local que global) afin de coupler plus étroitement les processus de perception et d'action.

3.2 Utilisation de la redondance en asservissement visuel

Pour définir une tâche en asservissement visuel, il n'est pas toujours nécessaire de disposer d'une quantité importante d'informations sur l'environnement. Schématiquement, seules les positions 2D des primitives visuelles utilisées dans la loi de commande sont nécessaires (auxquelles on devra cependant ajouter un minimum d'information 3D pour estimer la matrice d'interaction). L'asservissement visuel est donc une approche très locale et il est difficile de prévoir *a priori* à la fois le comportement 3D du manipulateur et la trajectoire 2D des primitives visuelles dans l'image. Cependant, si la commande calculée emmène la caméra ou le manipulateur dans une configuration indésirable, la tâche d'asservissement sera en échec. Il est donc important de construire des lois de commande capables de prendre en compte ces configurations indésirables.

Pour cela, des techniques de planification de trajectoire dans l'image reposant sur la méthode des champs de potentiel ont été développées [148]. Les fonctions de potentiel choisies sont définies dans l'espace opérationnel du robot et contraignent la trajectoire 3D du robot à suivre une ligne droite dans cet espace ce qui revêt un grand intérêt pratique. Il est ensuite possible d'introduire des contraintes dans les fonctions de potentiel sous forme de pôles répulsifs afin de faire dévier le robot de sa trajectoire nominale. Cette approche est intéressante puisqu'elle permet l'introduction de contraintes même si les six degrés de liberté sont contraints par la tâche visuelle (en fait, seule la trajectoire du manipulateur est modifiée pendant l'asservissement, et cette approche suppose évidemment que la position finale spécifiée est atteignable). Des techniques similaires, reposant sur des fonctions de navigation, ont aussi été proposées dans [49].

Si tous les degrés de liberté ne sont pas contraints par la tâche visuelle, l'utilisation de la redondance (voir section 2.2) est une solution intéressante permettant de pallier l'absence de planification résultant du calcul en ligne de la commande. Les contraintes sont introduites directement dans la loi de commande sous la forme d'une fonction de coût à optimiser. Si ce principe de redondance est bien exploité (par exemple en utilisant l'approche fonction de tâche), la tâche secondaire qui modélise les contraintes sur le système n'aura aucun effet sur la tâche principale. Dans le cas de l'asservissement visuel, cette so-

lution a été principalement utilisée pour réaliser des tâches de suivi de trajectoire. Il est cependant possible d'utiliser cette approche pour éviter des configurations indésirables. Comme nous l'avons indiqué dans le chapitre précédent, ceci est réalisé en définissant une fonction de coût à minimiser reposant sur une mesure du risque d'apparition de ces configurations. Ces tâches secondaires peuvent servir à éviter des configurations indésirables de la caméra ou du manipulateur par rapport à son environnement (évitement d'obstacles, par exemple) ou par rapport à ses singularités (internes ou externes – les butées articulaires –). Les contraintes peuvent aussi s'exprimer directement dans l'espace image. Ainsi les tâches secondaires peuvent servir à éviter des configurations indésirables (occultations) ou encore atteindre (ou maintenir) des configurations optimales dans l'image (champ de vue, gestion du flou). La tâche secondaire est alors elle-même une tâche visuelle.

Dans le cadre de nos travaux, nous avons donc considéré différents types de tâches que ce soit dans un contexte d'asservissement visuel pour la robotique ou, comme nous le verrons dans le chapitre 4, dans le contexte de l'animation de caméras et d'humanoïdes de synthèse dans des mondes virtuels :

- suivi de trajectoire [31]
- évitement des butées articulaires et des singularités *[135, 32]*,
- évitements d'obstacles avec *[138]* ou sans connaissance 3D *[139]*,
- positionnement en utilisant conjointement une caméra embarquée et une caméra déportée *[66]*,
- introduction de contraintes visuelles comme l'évitement des occultations ou le maintien de plusieurs objets dans le champ de vision de la caméra *[139]*,
- exploration pour la reconstruction de scènes 3D *[130]*.

Dans ce document nous présenterons uniquement la solution originale que nous avons proposée pour résoudre le problème de l'évitement des butées articulaires et nous évoquerons le cas particulier des contraintes exprimées dans l'espace image au travers de l'évitement des occultations.

3.2.1 Évitement des butées articulaires

Pour toutes les tâches robotiques, et notamment dans le cas de l'asservissement visuel, un problème très important est d'essayer d'éviter les butées articulaires et les singularités internes du robot. Les premières sont des limites physiques à l'extension de l'espace opérationnel du robot, tandis que les deuxièmes sont des configurations particulières où le robot perd localement des degrés de liberté ce qui rend impossible la génération de certains mouvements. Si l'on ne prend pas en compte ces limites, les tâches robotiques ne peuvent pas être réalisées lorsque les lois de commande produites amènent le robot en singularité ou sur ses butées articulaires. Nous avons réalisé une étude sur ce sujet qui nous a amené à proposer une solution originale (et générale) au problème de la gestion des butées articulaires *[32]*. La solution décrite dans *[135]* pour prendre en compte les singularités étant plus classique nous n'y reviendrons pas dans ce document.

Approche par projection de gradient. Classiquement le problème des butées articulaires est traité en utilisant la redondance des manipulateurs. Les degrés de liberté non contraints par la tâche principale sont utilisés afin de s'éloigner du voisinage des bu-

tées [114, 173]. Pour ce faire, une fonction de coût, reflétant la distance du robot à ces butées et définissant donc une notion de risque, est définie. Afin d'appliquer cette approche, la fonction de coût à minimiser doit être telle qu'elle atteigne sa valeur maximale quand le manipulateur arrive à proximité d'une butée articulaire. Ces fonctions de coût sont ensuite intégrées dans une tâche secondaire et régulées conjointement à la tâche visuelle. Dans [156], la tâche finale est une combinaison linéaire de la tâche visuelle et de la tâche secondaire ce qui pénalise bien les mouvements amenant le robot près de ses butées mais les mouvements générés produisent aussi d'importantes perturbations dans le processus d'asservissement visuel car ils ne sont généralement pas compatibles avec la régulation à zéro des primitives visuelles sélectionnées. La solution, classique, que nous avons retenue dans un premier temps repose sur le formalisme de la redondance décrit succinctement dans la section 2.2. Le gradient de la fonction de coût projeté sur le noyau de la tâche principale [114, 173] est utilisé pour générer les mouvements nécessaires à la minimisation de la fonction de coût h_s. Ce processus permet d'assurer que le processus d'évitement des butées articulaires n'a aucun effet sur la tâche visuelle.

En notant $\bar{\mathbf{q}}$ les butées basses et hautes qui ne doivent jamais être dépassées, il est possible de définir une fonction de coût h_s reflétant le comportement souhaité et représentée sur la figure 3.5. Les seuils d'activation $\tilde{\mathbf{q}}_{min}$ et $\tilde{\mathbf{q}}_{max}$ définissent les zones de l'espace articulaire où le processus d'évitement doit être considéré. h_s est nulle entre les seuils d'activation et croît de manière quadratique dans la zone critique à l'approche des butées.

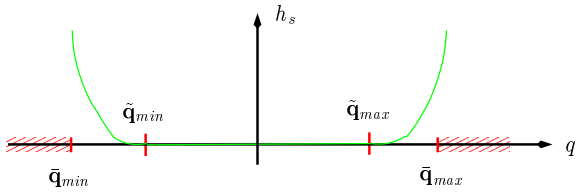


FIG. *3.5: Évitement des butées articulaires par une approche classique de projection de gradient : évolution de la fonction de coût h_s en fonction de la position articulaire.*

L'amplitude des mouvements du manipulateur dus à la tâche secondaire (voir équation (2.10) et figure 3.6) est gérée par un unique paramètre dont le réglage est extrêmement critique. Trop faible, la tâche secondaire ne sera pas suffisante pour éviter les butées articulaires (voir l'illustration sur la figure 3.6). Trop fort, des vitesses trop importantes peuvent *a contrario* être générées et la loi de commande résultante sera instable. Une valeur minimale de ce paramètre peut être fixée, mais cette solution n'assure pas que d'autres axes ne se rapprochent pas de butées *[32]*. Ces remarques sur le réglage de l'amplitude de la tâche secondaire sont générales à tous processus d'évitement (occultation, obstacle, etc) par des approches de projection de gradient et ne se limitent donc pas à l'évitement des butées articulaires.

Une approche optimale itérative. Pour résoudre ce problème de gain, nous avons proposé une méthode consistant à générer automatiquement des mouvements de caméra compatibles avec la tâche principale en résolvant simplement et itérativement un système linéaire d'équations. Cette nouvelle approche est plus efficace que l'approche classique par

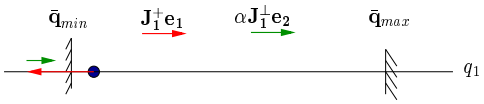


FIG. *3.6: Influence du gain α sur l'efficacité des approches par projection de gradient : si α est trop petit, les mouvements générés par $\mathbf{e_2}$ peuvent s'avérer inefficaces.*

projection de gradient. Elle évite des mouvements inutiles et garantit l'évitement des butées articulaires.

Une solution pour réaliser le processus d'évitement des butées est donc de couper tous les mouvements du manipulateur sur les axes qui sont en situation critique (c'est-à-dire entre $\tilde{\mathbf{q}}$ et $\bar{\mathbf{q}}$ et se rapprochant de $\bar{\mathbf{q}}$). En considérant que l'axe $\mathbf{q}_k$ est l'un de ces axes, on souhaite donc calculer une vitesse optimale $\dot{\mathbf{q}}_k = 0$. L'idée est de définir non plus un gain global α gérant l'amplitude des mouvements sur tous les axes, mais de calculer un vecteur de gains optimaux sur les seuls axes critiques (ou susceptible de le devenir).

Une alternative à l'équation (2.10) pour utiliser la redondance est donnée par :

$$\mathbf{e} = \mathbf{J_1^+}\mathbf{e_1} + \sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_i \mathbf{E}_{\bullet i} \tag{3.1}$$

où $n_\mathbf{a}$ est la dimension du noyau de $\mathbf{J_1}$ et la composante $\sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_i \mathbf{E}_{\bullet i}$ de cette équation définit les mouvements du robot devant assurer la contrainte d'évitement des butées articulaires ($\mathbf{E}$ est une base du noyau de $\mathbf{J_1}$ et $\mathbf{a}$ est le vecteur de gain que l'on cherche à déterminer). Comme dans le cas précédent les mouvements générés par cette composante sont évidemment totalement compatibles avec la tâche principale grâce au choix de $\mathbf{E}$.

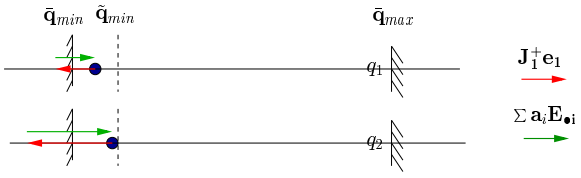


FIG. *3.7: Nouvel algorithme : les mouvements sur les axes en zones critiques sont tous stoppés grâce au calcul d'un vecteur de gains optimaux $\mathbf{a}$.*

Si l'on fait l'hypothèse que plusieurs axes sont en zone critique, il est nécessaire de calculer un vecteur $\mathbf{a}$ qui permette de stopper complètement le mouvement sur ces axes (voir figure 3.7). Si plusieurs axes sont en situation critique, on peut définir un système d'équations linéaires (à partir de (3.1)) en imposant une vitesse nulle sur ces axes.

La système linéaire ainsi formé (voir *[32]* pour plus de détails sur la formation de ce système) permet de calculer le vecteur de gain recherché $\mathbf{a}$. Deux cas peuvent se présenter lors de la résolution de ce système :

– si le nombre d'axes en situation critique est supérieur au nombre de degrés de liberté redondants, il n'y a rien a faire et le succès de l'évitement ne peut évidemment être assuré en utilisant la redondance.

– si le nombre d'axes en situation critique est inférieur ou égal au nombre de degrés de liberté redondants alors le système présente une ou plusieurs solutions.

Dans ce dernier cas, la solution trouvée implique que tout axe en situation critique a une vitesse nulle. Le problème peut ainsi sembler résolu. Dans la grande majorité des cas cette solution est satisfaisante et résout un grande nombre des problèmes soulevés par les approches classiques de projection de gradient. La loi de commande résultante peut cependant dans certain cas, générer des mouvements qui vont faire entrer de nouveaux axes en situation critique. Ce cas est illustré sur la figure 3.8. Au départ seul l'axe $\mathbf{q_1}$ est critique. La loi de commande générée après le calcul de $\mathbf{a}_0^*$ stoppe donc les mouvements sur cet axe en générant un mouvement "inverse" (flèche verte) à celui généré par la tâche principale (flèche rouge). Considérons maintenant le cas de l'axe $\mathbf{q_2}$: celui-ci n'était pas initialement en zone critique, mais le mouvement généré par la tâche secondaire (en vert), fait que cet axe devient aussi critique.

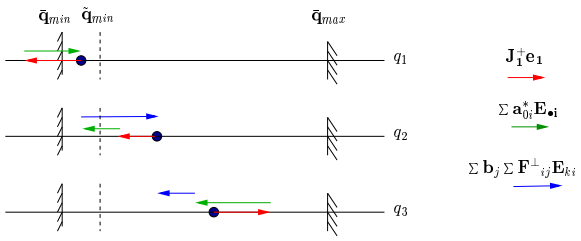


FIG. *3.8: Nouvel algorithme dans sa version itérative : aucun mouvement n'est autorisé pour les axes en zone critique et pour ceux susceptible d'y entrer. Si une solution existe au problème d'évitement de butée (i.e., si il y a des degrés de liberté disponibles), elle sera trouvée.*

Cette situation peut cependant être traitée si des degrés de liberté sont encore disponibles c'est-à-dire si le système a potentiellement plusieurs solutions. La solution $\mathbf{a}_0^*$ calculée n'est en effet qu'une des solutions de ce système (celle de norme minimale fournie par la pseudo-inverse utilisée pour résoudre le système linéaire). Toute autre inverse généralisée peut *a priori* être utilisée et là encore il existe une infinité de solutions au problème. Pour calculer les vitesses articulaires adéquates, comme dans le cas précédent, il est nécessaire de construire un système linéaire connaissant $\mathbf{a}_0^*$ et les nouveaux axes entrant en zones critiques. Ce processus, totalement décrit dans *[32]*, peut être itéré tant que des axes sont susceptibles d'entrer dans la zone critique en raison des mouvements générés par la tâche secondaire et tant que des degrés de liberté sont disponibles.

Cette approche originale assure que les mouvements sur les axes en situation critique seront stoppés et qu'aucun autre axe n'entrera dans la zone critique et ce tant que des degrés de liberté seront disponibles. Cette approche peut aussi être modifiée pour intégrer un processus d'éloignement des butées articulaires. Comme dans le cas classique des approches par projection de gradient, une tâche secondaire définie comme le gradient d'une fonction de coût peut être considérée. Un composante $\mathbf{J}_1^{\perp}\mathbf{e_2}$ (flèche bleu sur la figure 3.9) peut être ajoutée à la fonction de tâche (3.1) éloignant le manipulateur de la zone critique. Dans ce cas le réglage du paramètre gérant l'amplitude de cette tâche n'est plus critique puisqu'il ne participe plus à l'évitement mais seulement à l'éloignement des butées. Cependant on observe des discontinuités dans la commande dès que de nouveaux axes entrent en zone

critique (ces discontinuités inévitables sont en pratique peu gênantes et peuvent en grande partie être atténuées en modifiant légèrement la loi de commande [32]).

FIG. *3.9: Évitement des butées par notre approche itérative avec introduction d'une tâche secondaire.*

Les résultats de ces différentes approches ainsi que des comparaisons avec d'autre méthodes [114, 26]*[135]* sont présentés dans *[32]*. Ces techniques ont été utilisés en vision par ordinateur pour étendre les mouvements d'une caméra dans un processus de reconstruction 3D par vision active *[132]* et en animation pour gérer les butées mécaniques limitant les mouvements d'un humanoïde de synthèse *[46]* (voir section 4.3). Précisons finalement que la solution originale que nous venons d'évoquer dépasse largement le cadre de l'asservissement visuel et peut évidemment être utilisée pour l'évitement des butées articulaires dans d'autres contextes en robotique. Par ailleurs l'évitement des butées articulaires n'est sans doute pas le seul problème que l'on peut tenter de résoudre par cette nouvelle approche : on peut penser à la limitation des vitesses ou des accélérations angulaires, ou même éventuellement à la limitation des forces exercées sur l'effecteur.

Finalement quand les six degrés de liberté sont contraints par la tâche visuelle, ces techniques ne sont plus utilisables. Comme nous l'avons évoqué en introduction de cette section, une autre approche décrite dans [148] couple l'asservissement visuel avec les techniques de champ de potentiel, ce qui permet d'introduire des contraintes dans la trajectoire du robot et ce même si tous les degrés de liberté sont contraints. Une contrainte sur les butées articulaires a ainsi été proposée. La formulation du champ répulsif associée est relativement proche de la fonction de coût définie dans [114, 173]*[135]* et pose donc les mêmes problèmes de paramétrage.

3.2.2 Introduction de contraintes exprimées dans l'image

Les contraintes que l'on peut imposer sur la trajectoire du manipulateur ne s'expriment pas nécessairement, comme dans le cas précédent, à partir des paramètres internes du manipulateur ou le suivi d'une trajectoire apprise (et qui donc ne dépendent pas de la perception de l'environnement) mais peuvent aussi s'exprimer sous la forme de contraintes dans l'espace du capteur. Pour réaliser certaines tâches, ces contraintes peuvent dépendre de mesures faites à partir de capteurs extéroceptifs comme des télémètres laser, des caméras, des capteurs de force, etc. Dans le cas où ce capteur est une caméra, la tâche secondaire peut être une tâche visuelle si la contrainte s'exprime directement dans l'image. C'est par exemple le cas pour éviter des occultations, pour contraindre un objet à rester dans le champ de vue, pour gérer le flou, ou imposer des contraintes sur la résolution ou la "*resolvability*" [157]. D'autres problèmes comme la coopération multi-capteurs (caméra

embarquée – caméra déportée) peuvent aussi se résoudre via l'introduction de tâches secondaires visuelles *[139][66]*.

L'exemple de l'évitement des occultations est, dans ce sens, assez parlant. L'objectif est d'éviter qu'un objet mobile, dont la trajectoire est inconnue, vienne se situer entre la caméra et l'objet d'intérêt sur lequel la caméra est asservie (voir figure 3.10). En supposant que l'objet "occultant" passera bien entre la caméra et la cible (et non derrière l'objet), on se ramène à un problème uniquement 2D. Il faut éviter que la projection $\mathbf{o}$ de l'objet sur le plan image se rapproche de la projection $\mathbf{t}$ de la cible. Ce problème pourrait très bien se définir entièrement comme une tâche d'asservissement visuel (on désire voir $\mathbf{o}$ et $\mathbf{t}$ à telle et telle position dans l'image mais la contrainte, exprimée sous cette forme, est beaucoup trop forte).

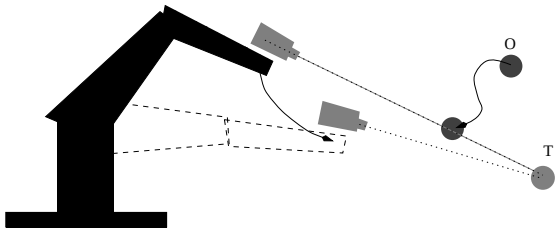


FIG. *3.10: Exemple d'une contrainte exprimée dans l'image : l'évitement d'occultations. La caméra doit rester focalisée sur l'objet T en évitant les occultations potentielles provoquées par O*

Il faut alors définir une fonction de coût h_s qui atteint son maximum lorsque l'objet est occulté, c'est-à-dire quand la distance dans l'image entre $\mathbf{o}$ et $\mathbf{t}$ est nulle. On cherche alors à maximiser la distance dans l'image entre ces deux objets. Pour cela, h_s peut, par exemple, être définie par *[139]* :

$$h_s = \frac{1}{2}\alpha e^{-\beta\|\mathbf{t}-\mathbf{o}\|^2} \tag{3.2}$$

où α est un scalaire qui règle l'amplitude de la réaction de la caméra (plus α est élevé et plus la vitesse de la caméra sera importante) et où β permet de définir le moment à partir duquel le mouvement secondaire va se déclencher en fonction de la distance dans l'image entre l'objet d'intérêt et l'objet occultant (plus β est important et plus l'objet peut se rapprocher de l'objet d'intérêt avant que la réaction ne se produise). La tâche secondaire $\mathbf{e_2}$ dérivée à partir de la fonction de coût h_s est une tâche visuelle. Cette tâche a été aussi utilisée dans un contexte de contrôle de caméra dans un environnement virtuel (voir section 4.1.2).

3.3 Asservissement visuel robuste aux mesures aberrantes

Si l'asservissement visuel est très efficace pour réaliser des tâches de positionnement, il apparaît cependant que la précision de positionnement est très sensible aux erreurs inhérentes au processus d'extraction des données. L'efficacité de l'asservissement visuel dépend

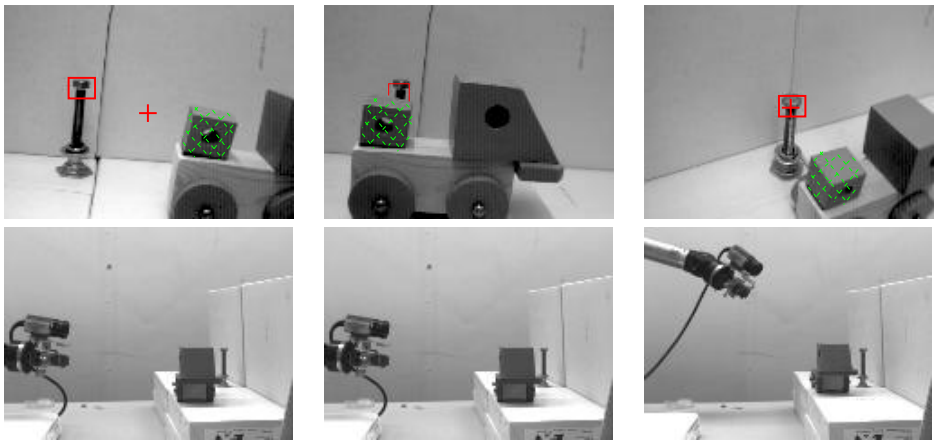


FIG. *3.11: Exemple d'une contrainte exprimée dans l'image : l'évitement d'occultations. La figure (a) montre le positionnement de la cible (rouge) et de l'objet occultant (vert) avant le début de la tâche. La figure (b) montre le résultat du positionnement si rien n'est fait pour éviter l'occultation (ici partielle) et la figure (c) montre le résultat du positionnement en considérant une tâche secondaire d'évitement d'occultations. Les images du bas montrent les vues extérieures respectives de la scène et du manipulateur (robot Zebra zero de l'université de Yale)*

en effet de la précision de localisation de cette information visuelle mais aussi de la précision de l'appariement entre les valeurs courante et désirée de cette information. Si la mise en correspondance entre les informations visuelles est entachée d'erreur ou si l'estimation de la valeur de $\mathbf{s}$ est imprécise, la précision de la tâche de positionnement sera imprécise, voire même, dans certains cas, l'asservissement sera un échec.

Traditionnellement, la robustesse d'une loi de commande est définie par : "*stability results which remain true in the presence of* ***modeling errors*** *or certain classes of* ***disturbance***" [173]. Deux solutions peuvent donc être exhibées pour assurer la robustesse de la loi de commande : la première est de créer un modèle le plus précis possible du système considéré (perturbations potentielles comprises) et la seconde est de traiter (limiter) au mieux les perturbations en travaillant directement sur la commande. Dans le premier cas, il est raisonnable de penser qu'une modélisation et une estimation correcte de l'ensemble des paramètres intrinsèques du système permettent d'améliorer les résultats. En asservissement visuel, ce type d'approche a conduit à modéliser la caméra par un modèle de projection perspective, à disposer d'une formulation analytique de la matrice d'interaction [28] et à estimer en ligne l'information de profondeur présente dans cette matrice [146, 73, 31, 190], etc. D'autres sources d'erreurs proviennent du bruit dans l'extraction des indices visuels, ou d'erreurs de suivi voire d'importantes erreurs de mise en correspondance entre primitives courantes et désirées. La prise en compte de ces erreurs se fait le plus souvent en aval de la loi de commande, c'est-à-dire au niveau de l'extraction des indices visuels (voir figure 3.12a) : amélioration de la qualité des algorithmes de suivis [193] ou sélection de primitives particulières [161], fusion d'informations redondantes (par des approches de vote ou de consensus [101]).

Les solutions mentionnées dans le paragraphe précédent sont des solutions partielles

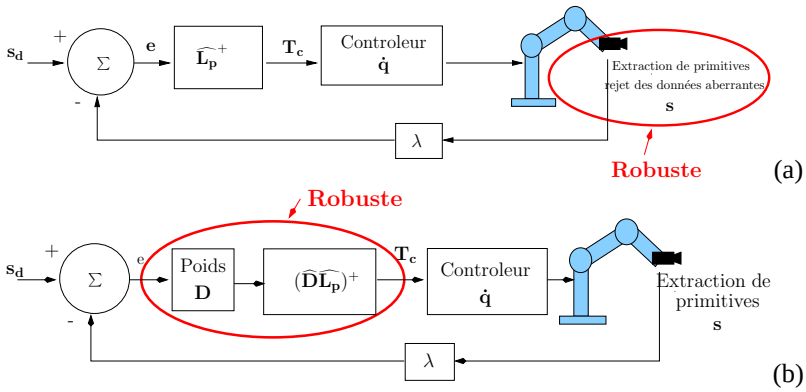


FIG. *3.12: (a) Asservissement visuel robuste "classique" : le rejet des données aberrantes se fait dans l'extraction des données, (b) Nouvelle loi de commande : le rejet des données aberrantes se fait dans la loi de commande.*

pouvant prendre en compte certains types d'erreurs bien définis. Une séquence d'images acquises à la cadence vidéo est cependant une source quasi infinie d'erreurs qu'il est impossible de caractériser et de traiter de manière exhaustive. Ceci inclut les problèmes dus au mouvement plus ou moins rapide des objets, aux occultations éventuellement multiples, aux changements d'illumination, etc. Il semble évident qu'établir un catalogue analytique de toutes les sources de perturbations possibles et de proposer une solution pour traiter chacune d'entre elles est un travail complexe voire impossible à réaliser. Nous avons donc décidé d'essayer de limiter l'effet des perturbations potentielles en modifiant la loi de commande. Nous avons en effet considéré le problème de l'asservissement visuel robuste en introduisant directement dans la loi de commande des estimateurs robustes permettant de quantifier la confiance dans chacune des informations visuelles et, si nécessaire, de les rejeter (voir figure 3.12b). L'incertitude sur chaque primitive est donc modélisée statistiquement, ce qui permet de prendre en compte tout type de perturbations dans l'extraction des données.

Dans la littérature portant sur les statistiques ou la vision par ordinateur [1], différentes approches ont été proposées pour considérer la détection et le traitement des sources de perturbation : méthode Ransac, algorithme LMedS, estimateur robuste,... Les approches de type Ransac ou LMedS se prêtant mal à une intégration dans une loi de commande d'asservissement visuel (autrement que pour l'initialisation), nous avons proposé une méthode reposant principalement sur l'utilisation des M-estimateurs même si l'algorithme LMedS peut nous fournir une initialisation intéressante. Les M-estimateurs peuvent être considérés comme une formulation générale d'un estimateur au maximum de vraisemblance [86]. Ils sont plus généraux car ils permettent l'utilisation de différentes fonctions de minimisation qui ne correspondent pas nécessairement à une distribution normale des données. Un grand nombre de fonctions robustes ont été proposées dans la littérature qui permettent de considérer comme peu vraisemblables des mesures incertaines et, dans certains cas, de

[1] Nous reviendrons plus longuement sur ces techniques d'estimation robuste en vision par ordinateur dans le chapitre 5.

les rejeter complètement. La loi de commande intégrant les M-estimateurs se formule de manière similaire à un algorithme d'estimation reposant sur les moindres carrés pondérés itérés (*Iteratively Re-weighted Least Square*). Afin d'améliorer la précision de détection des données aberrantes, la valeur de la variance du bruit de mesure (pour les données non aberrantes) est estimée au cours de la minimisation. Pour cette estimation, puisque les données peuvent contenir des "*outliers*", nous utilisons la valeur médiane de la déviation absolue (Median Absolute deviation ou MAD) qui représente un estimateur robuste de l'écart type du bruit de mesure [86].

Nouvelle loi de commande. Nous considérons la tâche générique qui consiste à déplacer une caméra pour observer un objet à une position donnée dans l'image. Ceci est accompli en minimisant l'erreur Δ entre un état désiré des primitives dans l'image $\mathbf{s}^*$ et leur état courant $\mathbf{s}$ (voir équation (2.1)).

En asservissement visuel, la loi de commande qui réalise la minimisation de Δ est traitée habituellement par une approche aux moindres carrés [28, 87]. Cependant, s'il y a des données aberrantes, la réalisation de la tâche sera en échec et une prise en compte explicite de ce problème est nécessaire. Comme évoqué dans le paragraphe précédent, notre approche repose principalement sur l'utilisation des M-estimateurs. La fonction à minimiser est donc modifiée afin de réduire la sensibilité aux données aberrantes. L'erreur à minimiser est alors donnée par :

$$\Delta_{\mathcal{R}} = \sum_{i=1}^{N} \rho\big(s_i(\mathbf{r}) - s_i^*\big)^2, \tag{3.3}$$

où $\rho(u)$ est une fonction robuste [86].

De façon similaire au problème des moindres carrés pondérés itérés, nous introduisons dans la loi de commande une matrice de pondération, où les poids reflètent la confiance dans chaque primitive visuelle.

Loi de commande robuste. En asservissement visuel classique, une fonction de tâche $\mathbf{e}$ permettant de minimiser l'erreur $\|\mathbf{s} - \mathbf{s}^*\|$ est définie par la relation (2.4). Nous avons proposé une nouvelle loi de commande $\mathbf{e}$ qui assure une minimisation robuste de Δ définie par l'équation (3.3). Elle est définie par la relation[2] :

$$\mathbf{e} = \mathbf{D}\left(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\right), \tag{3.4}$$

où $\mathbf{D} = diag\big(w_1, \ldots, w_k\big)$ est une matrice diagonale. Le calcul du poids w_i associé à chaque information visuelle représente la confiance que l'on a dans chacune des informations visuelles. Le calcul de ces poids est un point fondamental de cet algorithme et sera décrit dans le paragraphe suivant. Sans entrer dans les détails (voir *[39]*), on obtient une loi de commande donnée par :

$$\mathbf{v} = -\lambda(\widehat{\mathbf{D}}\widehat{\mathbf{L}_{\mathbf{s}}})^{+}\mathbf{D}\big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big). \tag{3.5}$$

[2]Par souci de simplicité nous n'avons pas considéré ici la matrice de combinaison $\mathbf{C}$ utilisé dans l'équation (2.4). On ne peut donc plus formellement parler de fonction de tâche pour l'équation (3.4). La dérivation complète de la loi de commande est cependant donnée dans *[39]*.

où un modèle ou une approximation $\widehat{\mathbf{L_s}}$ de $\mathbf{L_s}$ sont utilisés (un modèle $\widehat{\mathbf{D}}$ de $\mathbf{D}$ peut aussi être considéré). La convergence et la stabilité sont des questions importantes lorsque l'on applique une telle loi de commande. Les résultats de stabilité sont équivalents à ceux obtenus pour un asservissement visuel 2D classique sous l'hypothèse que les données aberrantes sont correctement rejetées *[39]*.

Précisons qu'il est bien évidemment nécessaire de s'assurer qu'un nombre suffisant d'informations visuelles ne sera pas rejeté par les estimateurs robustes afin que $\mathbf{DL_s}$ soit toujours de rang plein (6 pour contrôler les 6 degrés de liberté du robot). La pseudo-inverse $(\mathbf{DL_s})^+$ étant calculée via une décomposition SVD, on a facilement accès au rang de la matrice $\mathbf{DL_s}$ ce qui permet de vérifier qu'elle est de rang plein. Finalement, comme notre approche repose sur la redondance d'informations, il est impossible de la considérer dans le cadre des commandes 2D 1/2 [122] ou 3D puisqu'un nombre minima d'informations visuelles est utilisé dans ce type d'approche.

Calcul du degré de confiance. Parmi les diverses fonctions robustes $\rho(.)$, nous avons retenu la fonction de Tukey dont la fonction d'influence (utile pour le calcul de poids) rejette complètement les données aberrantes et leur donne un poids nul [86]. Il est en effet souhaitable que les données aberrantes n'aient ***aucun*** effet sur le mouvement de la caméra (ce n'est pas le cas avec d'autres fonctions robustes comme les fonctions de Huber, Cauchy ou Geman). Des erreurs dans l'image, même petites, peuvent en effet entraîner un écart très important dans la précision du positionnement final. Cette fonction d'influence est donnée par :

$$\psi(u) = \begin{cases} u(C^2 - u^2)^2, & \text{si } |u| \leq C \\ 0, & \text{sinon} \end{cases} \tag{3.6}$$

où le facteur de proportionnalité pour la fonction de Tukey est $C = 4,6851$ [86]. Ce facteur représente une efficacité de 95% dans le cas du bruit Gaussien.

Les poids w_i, éléments de la matrice $\mathbf{D}$, reflètent la confiance en chaque primitive et sont définis par [86] :

$$w_i = \frac{\psi(\delta_i/\sigma)}{\delta_i/\sigma} \tag{3.7}$$

où δ_i est le résidu normal donné par $\delta_i = \Delta_i - \text{med}\ \Delta$ (med Δ correspond à la valeur médiane des résidus). Le paramètre σ, qui représente la valeur de l'écart type du bruit sur les "bonnes" mesures, peut varier énormément au cours du processus de minimisation. σ est souvent traitée comme une variable d'ajustement qui est choisie manuellement en fonction d'une application particulière. Dans notre cas, afin d'améliorer la précision de détection des données aberrantes, la valeur de σ est estimée parallèlement à la minimisation de l'erreur en utilisant une statistique robuste (le Mad pour *Median Absolute Deviation*, voir *[39]* pour plus de détails).

Initialisation des poids. L'expérience montre que quand l'erreur $\mathbf{s} - \mathbf{s}^*$ est importante (typiquement au début d'une tâche de positionnement), les erreurs dues à la présence de données aberrantes ne sont pas nécessairement statistiquement significatives (du moins au

sens des M-estimateurs). Dans ce cas les poids ne seront pas égaux à zéro et la tâche s'en retrouvera fortement perturbée. Il est donc important de pouvoir détecter la présence de ces données aberrantes dès l'initialisation du processus d'asservissement, pour un coût calculatoire éventuellement plus important, et d'initialiser correctement les poids.

Pour cela nous avons utilisé l'algorithme LMedS (least-median-of-squares [169]) qui, dans notre cas, consiste à déterminer la valeur de la vitesse $\mathbf{v}$ de la caméra qui minimise le critère suivant :

$$\widehat{\mathbf{v}} = \underset{\mathbf{v}}{\operatorname{argmin}} \underset{i=1,\ldots n}{\operatorname{med}} \left(\mathbf{L}_{s_i}\mathbf{v} + (s_i - s_i^*)\right)^2 \tag{3.8}$$

où $\mathbf{L}_{s_i}$ et s_i sont respectivement la i-ème ligne de la matrice d'interaction $\mathbf{L_s}$ et du vecteur $\mathbf{s}$.

Contrairement aux M-estimateurs, la minimisation de ce critère ne peut se réduire à la résolution d'un système linéaire pondéré et ne peut être résolue analytiquement. On doit donc explorer l'ensemble des estimations potentiellement générées par les données (ensemble qui peut vite devenir énorme mais dont la taille peut se réduire en ayant recours à des tirages de Monte Carlo). Si cette approche pourrait *a priori* fournir une commande au système, il n'est pas réaliste de l'utiliser avec cet objectif (les temps de calcul n'étant pas compatibles avec la cadence vidéo). Elle fournit cependant une initialisation binaire (0 ou 1) des poids très robuste *[137]*.

Résultats. Les résultats obtenus ont montré l'efficacité d'une telle approche (comme le montrent les résultats de la figure 3.13. Des résultats plus complets sont données dans *[39][137]*). Il reste que l'utilisation d'une telle loi de commande robuste n'est pas incompatible, loin de là, avec un processus efficace d'extraction des données. Une fusion des deux schémas de la figure 3.12 est non seulement possible mais souhaitable.

Bilan

L'asservissement visuel fait l'objet de recherches fructueuses depuis de très nombreuses années et est particulièrement intéressant par le spectre scientifique et applicatif très large qu'il recouvre. Il fournit une alternative au cycle classique de Perception → Décision → Action en liant plus étroitement les aspects de perception et d'action grâce à une intégration directe des mesures fournies par un système de vision dans des lois de commande en boucle fermée sur les informations visuelles extraites.

Nous avons commencé à utiliser ces techniques, pendant notre thèse, dans un contexte de vision active. L'objectif était alors de commander une caméra pour explorer et cartographier efficacement un environnement *a priori* inconnu. Devant les problèmes que posait l'intégration des ces techniques dans un système beaucoup plus complexe, il est vite apparu qu'il était nécessaire de modifier les lois de commande pour prendre en compte certaines contraintes sur le système. C'est à la suite de ce constat que nous avons réalisé nos premiers travaux sur la redondance. Nous verrons dans le chapitre suivant que ces travaux menés initialement dans un contexte robotique ont été largement utilisés dans le domaine de l'animation par ordinateur pour proposer de nouvelles métaphores de contrôle aux animateurs.

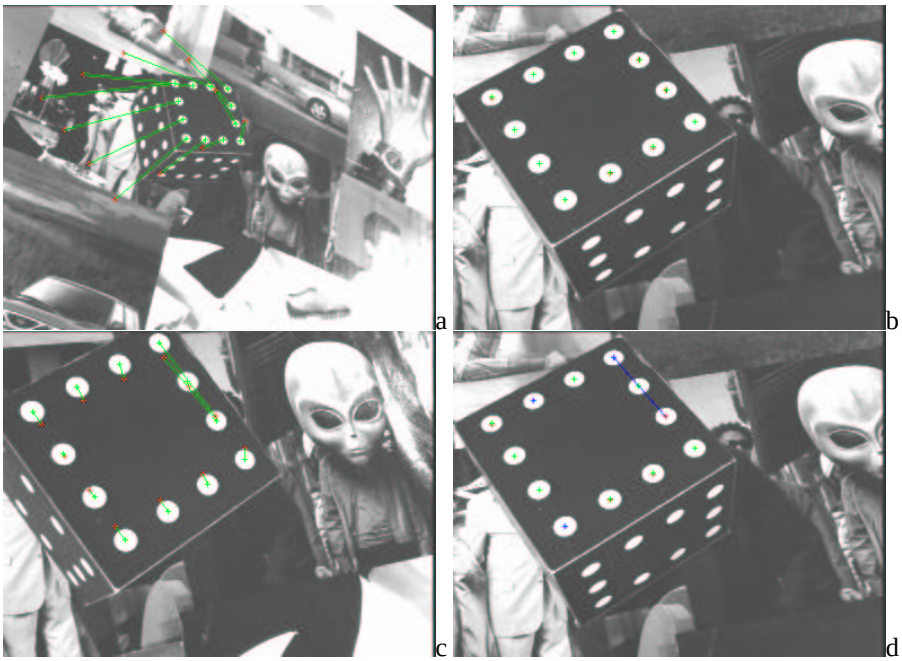


FIG. *3.13: Tâche classique de positionnement en considérant une loi de commande classique et une loi de commande robuste. L'image (a) montre l'image initiale acquise avant le début de la tâche. Les trois autres images correspondent aux images acquises à l'issue de la tâche de positionnement : (b) correspond à un positionnement reposant sur une loi de commande classique mais sans données aberrantes (expérience de référence), (c) reprend la même loi de commande classique mais la mise en correspondance entre points courants et désirés est faussée introduisant des données aberrantes (c'est un des cas d'erreurs possibles, mais d'autres cas son envisageables [39, 137]. Comme on peut s'y attendre la commande converge vers un minimum local, (d) considère la même expérience mais avec une loi de commande robuste. Malgré les données aberrantes, la tache de positionnement se déroule correctement.*

Un second problème récurrent rencontré est la sensibilité de l'asservissement visuel aux données aberrantes. Nous avons recherché et expérimenté avec succès une nouvelle approche dite d'asservissement visuel 2D robuste. Elle permet de considérer la gestion des données aberrantes directement au niveau de la loi de commande, ce qui présente l'avantage d'éviter un développement lourd d'algorithmes de traitement d'images robustes à tout type de perturbation. Cette loi de commande reposant sur la redondance de l'information est cependant malheureusement inexploitable en asservissement visuel 2D 1/2 ou 3D. Cette loi de commande a originellement été considérée dans un contexte de suivi comme nous le verrons dans la section 5.4.2 de ce document.

Précisons par ailleurs, que d'autres travaux relatifs à l'asservissement visuel dans un contexte robotique et portant sur la coopération multi-capteurs *[66]*, sur l'asservissement visuel déporté *[136]*, sur la projection de consignes ou encore la commande de manipulateurs non-

instrumentés[3] (en collaboration avec l'Ifremer à Toulon) ont aussi été réalisés *[136]* mais n'ont pas été décrits dans ce document.

[3]Un manipulateur non instrumenté est un manipulateur qui ne dispose pas de capteurs proprioceptifs renvoyant sa position articulaire. Il s'agissait en l'occurrence du bras Sherpa monté sur le ROV Victor 6000 de l'Ifremer

CHAPITRE 4

Animation d'une caméra virtuelle

Dans la section précédente nous avons considéré l'asservissement visuel dans son contexte naturel : la robotique. Après près de 15 ans de recherches dans ce domaine, il semblait légitime de se demander si l'asservissement visuel peut être considéré dans d'autres contextes. Nous avons retenu, dans un premier temps, le contexte de l'animation par ordinateur et la réalité virtuelle. Historiquement, les liens entre la robotique et l'animation par ordinateur sont nombreux (e.g., [71, 191, 82, 105]). Dans la plupart des cas les systèmes à animer sont considérés de manière similaire à des robots dont le comportement est simulé de façon réaliste. L'image synthétique qui en résulte n'est au départ souvent qu'une façon aisée de visualiser le comportement généré du système. Si le système à animer est bien modélisé et la tâche bien spécifiée, les mouvements générés peuvent s'avérer être extrêmement réalistes. Avec l'amélioration des techniques de visualisation, ces approches issues de la robotique peuvent être efficacement utilisées dans des applications où le réalisme de l'animation est fondamental (dans l'industrie cinématographique par exemple).

Le concepteur de mondes virtuels et synthétiques se base souvent sur les outils dont il dispose pour établir ce qu'il peut réaliser. Cette ingérence de l'outil, même si elle existe dans la plupart des arts ou des domaines de l'ingénierie, constitue un obstacle au processus de création et son atténuation représente l'un des buts actuels des recherches menées. D'une manière générale, ces outils permettent de modéliser des fonctionnalités précises, et d'automatiser certains processus de conception, pour rendre ce travail plus rapide et plus simple. Cette automatisation pose le problème du contrôle, c'est-à-dire l'estimation ou le calcul des paramètres permettant d'interagir avec le processus automatisé. Dans la continuité des travaux où robotique et animation sont étroitement mêlées, nous avons proposé des outils permettant de modéliser des éléments de base utilisables par de telles applications. Si en animation, l'image n'est souvent qu'un sous-produit (au demeurant fondamental) du processus de simulation du comportement du système, il est rare qu'elle soit utilisée pour contrôler automatiquement le système (voir cependant les travaux réalisés dans la lignée

de [197]). L'asservissement visuel est une technique de contrôle reposant sur la perception visuelle de l'environnement. Notre objectif a donc été de proposer des outils afin de contrôler l'animation d'entités virtuelles en fonction de l'image qu'elles ont du monde virtuel. Disposer d'un tel outil permet de contrôler les entités virtuelles à un niveau tâche, c'est-à-dire avec une abstraction du contrôle suffisamment importante pour être considérée comme proche du langage naturel.

4.1 Positionnement et navigation d'une caméra dans des espaces virtuels

Le contrôle d'une caméra dans un environnement virtuel soulève de nombreuses questions. De façon classique, la caméra doit non seulement pouvoir se positionner par rapport à son environnement, mais elle doit de plus être à même de réagir à des modifications de celui-ci. Concernant le premier point, même en considérant une connaissance complète de l'environnement, ce qui est généralement le cas en animation, réaliser une tâche de positionnement n'est pas comme on pourrait le croire un problème trivial (voir les commentaires de Blinn [16] à ce sujet). On peut, à ce stade, distinguer deux types d'exigences : d'une part la volonté du créateur de définir explicitement tous les déplacements de la caméra (auquel cas les différentes techniques de contrôle doivent lui permettre de réaliser des effets que l'on pourrait qualifier de cinématographiques), et d'autre part un désir de fournir à l'utilisateur un contrôle de haut niveau sur les éléments de l'environnement virtuel, par exemple poursuivre un objet tout en évitant son occultation par d'autres parties de la scène, ou se focaliser sur des composantes précises de la scène. Une difficulté est ici de prendre en compte les modifications de l'environnement.

Dans le domaine infographique, des approches référencées images ont aussi été considérées. La principale différence avec le domaine robotique est, même dans un contexte interactif, la connaissance exhaustive de l'état passé et de l'état actuel des objets du système (profondeur, vitesse,...). Ware et Osborne [201] proposent différentes métaphores pour décrire une caméra à six degrés de liberté ("*eyeball in hand*", "*scene in hand*" et "*flying vehicle*"). La plus intéressante de ces métaphores est "*eyeball in hand*", où la main désigne la position et l'œil l'orientation. Contrôler un tel objet n'est pas un problème trivial. Une solution est d'utiliser des périphériques comme une souris 3D ou un joystick à six degrés de liberté. Obtenir alors un mouvement fluide et réaliste nécessite un opérateur qualifié. La technique classique de paramétrisation de la caméra via trois vecteurs Lookat/Lookup/Vup permet de se focaliser simplement sur un point précis de l'environnement. Mais spécifier une tâche plus complexe et de plus haut niveau (par exemple "je veux garder cet arbre au centre de mon image et conserver le lapin bondissant autour de ce même arbre dans le quart gauche de l'image") s'avère difficilement modélisable avec cette technique. Des travaux dans ce sens furent menés par Blinn [16]. Cependant les résultats s'avèrent trop spécifiques et inadaptés aux problèmes multi-contraints. Différentes solutions pour résoudre l'introduction de contraintes ont été proposées tant en robotique [187, 48] qu'en animation [59]. Les solutions résultantes sont similaires : chaque contrainte est définie mathématiquement comme une fonction des paramètres de la caméra (position focale, zoom, etc.) devant être minimisée selon des méthodes déterministes (descente de gradient) ou stochastiques (recuit

simulé). Elles présentent cependant plusieurs défauts : de complexités souvent grandes, elles empêchent une implémentation temps-réel (recherche dans des espaces de dimension six) et nécessitent une optimisation à chaque itération. De plus, les problèmes multi-contraints induisent des fonctions de coût souvent très fortement non-linéaires nécessitant une initialisation adéquate et ne présentant pas nécessairement de solutions. Il reste qu'un grand nombre de fonctions de coût modélisant de nombreuses situations classiques dans le domaine cinématographique sont proposés dans [59].

4.1.1 Spécification des tâches dans l'espace image

Il apparaît cependant un fossé entre le niveau déclaratif de ce que l'on cherche à obtenir, et les moyens techniques pour y arriver. Les approches basées images (où la déclaration des contraintes s'effectue dans l'espace de ce que voit la caméra) sont des approches intéressantes dans le sens où la spécification des tâches est plus simple et se rapproche du formalisme utilisé en cinématographie [5]. Cependant celles-ci ne disposent la plupart du temps que de peu de flexibilité (résolution de problèmes dédiés) ou sont inadaptées pour les applications temps-réel.

Le contrôle explicite de la caméra à partir d'informations basées image a été étudié en animation dans [72]. Les auteurs proposent de positionner la caméra par rapport à des objets définis par des points virtuels statiques. Cette technique s'appuie sur une inversion locale de la matrice non-linéaire de transformation perspective. Une optimisation sous contraintes est alors utilisée pour calculer la vitesse de la caméra associée aux déplacements désirés des points virtuels dans l'image. Une autre formulation de ce problème a été établie dans [109]. On retrouve dans les deux cas une problématique et une formulation extrêmement proche voire équivalente à l'asservissement visuel. Cependant, la relation liant les mouvements de la caméra aux primitives visuelles n'est étudiée que pour des points, et aucune contrainte supplémentaire n'est introduite sur le mouvement de la caméra. Il n'en demeure pas moins que l'idée de l'utilisation de l'asservissement visuel est, volontairement ou non, présente dans ces travaux. Nous avons considéré dans le cadre de la thèse de Nicolas Courty une approche similaire reposant explicitement sur les techniques asservissement visuel [43].

Si l'utilisation d'une telle commande référencée capteurs est quasi indispensable en robotique pour définir le mouvement du robot, on peut directement se poser la question de l'intérêt de cette technique en synthèse d'images, car la plupart des informations sur l'environnement sont disponibles. L'intérêt se situe principalement au niveau des facilités de spécification offertes par une telle technique, et donc du niveau de l'abstraction du contrôle offert. En asservissement visuel 2D la tâche est spécifiée sous la forme d'une consigne visuelle à atteindre. Au travers des informations perçues par la caméra, des mouvements 3D sont générés dans le but de réaliser cette consigne exprimée dans un espace 2D. Ces "consignes visuelles" définissent donc une **abstraction du contrôle**. La figure 4.1 illustre l'exécution d'une telle tâche visuelle. Par rapport à une spécification classique de la position et de l'orientation dans l'espace 3D, la facilité de spécification de la tâche est accrue (même si, reconnaissons le, cette tâche est relativement simple). Cette méthode de contrôle s'adapte tout particulièrement bien à (au moins) deux types d'entités virtuelles : les caméras et un humanoïde de synthèse (ou toute autre entité décrite par une chaîne cinématique, voir section 4.3). Dans les deux cas, la notion de tâche visuelle prend un sens évident.

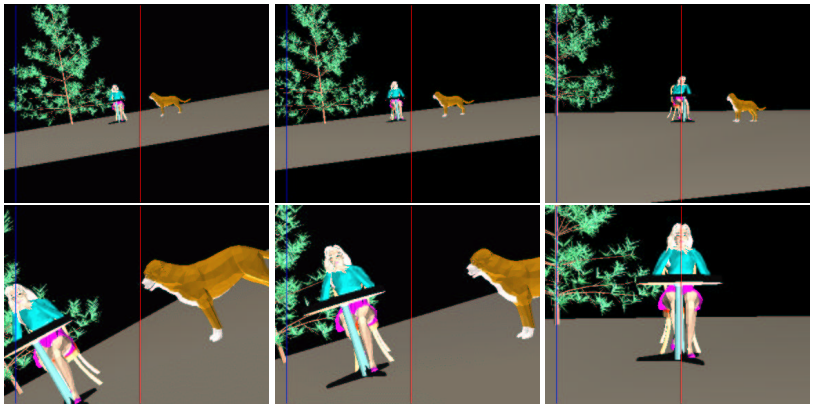


Fig. 4.1-a: *La tâche est ici spécifiée par "je veux voir l'arbre vertical sur la gauche (ligne bleue) et la jeune femme assise verticale au centre de l'écran (ligne rouge)". Si la spécification est simple, elle ne contraint que quatre degrés de liberté de la caméra. Il existe donc une infinité de positions de la caméra qui sont solution à ce problème (illustré ici pour deux positions initiale de la caméra)*

Fig. 4.1-b: *En spécifiant différemment la tâche "je veux voir l'arbre vertical sur la gauche (ligne bleu) et la tête de la jeune femme assise (schématisée par une sphère) dans le cercle rouge au centre de l'écran", cinq degrés de liberté sont alors contraints (en particulier la distance par rapport au personnage est imposée).*

FIG. *4.1: Positionnement d'une caméra virtuelle par asservissement visuel par une spécification dans l'image : Illustration de l'importance du choix de la tâche quant à la position finale de la caméra.*

Dans le cas d'une tâche de positionnement de caméra virtuelle dans un monde virtuel, la méthodologie présentée dans le chapitre 2 s'applique quasiment tel quel. Le vecteur d'informations visuelles $\mathbf{s}(\mathbf{r})$ est calculé en simulant la projection perspective d'une primitive 3D pour une position $\mathbf{r}$ de la caméra. La matrice d'interaction associée peut, dans ce cas, être calculée exactement (et non plus estimée) puisque l'on peut considérer qu'il n'y pas d'erreurs ni dans le calcul de $\mathbf{s}(\mathbf{r})$ ni dans celui de l'information Z nécessaire au calcul de $\mathbf{L_s}$. La tâche de l'animateur se limite donc à spécifier la consigne et la caméra se positionnera automatiquement.

La tâche de l'animateur est donc simplifiée puisque son rôle se limite à choisir les primitives visuelles et la consigne à atteindre dans l'image. Le choix des informations visuelles n'est cependant pas innocent. Si la tâche spécifiée ne contraint pas les six degrés de liberté de la caméra, il existe une infinité de positions minimisant l'erreur entre la position courante et la consigne. Pour deux positions initiales différentes, les positions finales de la caméra peuvent donc être (et seront sans doute) différentes. Ce point est illustré sur la figure 4.1, où pour deux positions initiales différentes une tâche de positionnement contraignant quatre (figure 4.1-a) et cinq (figure 4.1-b) degrés de liberté ont été spécifiées. Même si à l'issue du positionnement l'erreur dans l'image est nulle, la position finale de la caméra est très différente (en particulier dans l'exemple de la figure 4.1-a où la distance entre la caméra et la scène n'est pas contrainte par la tâche) ce qui du point de vue de l'animateur peut s'avérer gênant.

Dans le début de cette section, nous avons extrait un certain nombre de caractéristiques importantes pour un système d'animation de caméra : simplicité dans la spécification des tâches d'observation, adaptation à des modifications dynamiques de l'environnement, exécution en temps-réel. L'utilisation de l'asservissement visuel nous permet de répondre à la plupart de ces contraintes. L'originalité de ce système réside surtout dans le passage d'une spécification de la tâche en 2D à un mouvement en 3D. Mais cette approche n'est pas exempte de défauts : on ne dispose que de peu de contrôle sur la réalisation de la tâche, et donc sur les trajectoires 3D résultantes. Cette lacune peut rendre le système impropre aux besoins d'un animateur qui peut vouloir disposer d'un contrôle total sur les déplacements de la caméra.

4.1.2 Introduction de contraintes dans la commande

Une réponse partielle à ce besoin réside dans l'utilisation de la redondance. Les degrés de liberté non contraints par la tâche visuelle peuvent en effet permettre de compléter la spécification du comportement de la caméra en définissant des tâches secondaires (qui ne sont pas nécessairement spécifiées dans l'image).

L'utilisation de cette méthode permet de considérer des tâches simples de suivi d'objets ou de contraindre la trajectoire de la caméra tout en assurant une tâche de focalisation *[138][43]*. Dans le cas d'un suivi de trajectoire, celle-ci peut être définie par l'utilisateur ou de manière automatique si l'on considère un pré-travail de planification (voir figure 4.2). Ces exemples restent assez simples (ils ne considèrent qu'une cible isolée dans un environnement "vide"), mais l'utilisation de ce principe de redondance permet de résoudre des problèmes considérés comme non-triviaux dans le domaine de l'animation et du contrôle de la caméra. Il est possible d'envisager le cas où la caméra et les objets d'in-

térêt évoluent dans un environnement plus complexe, où d'autres objets (statiques ou non) peuvent gêner la perception de la scène. Dans cette optique, la redondance peut, comme en robotique, être utilisée pour résoudre deux problèmes récurrents en animation : l'évitement d'obstacle et l'évitement de l'occultation d'une cible d'intérêt par un autre objet de la scène. Ce ne sont là que deux exemples mais la méthodologie proposée peut s'adapter à d'autres types de problèmes.

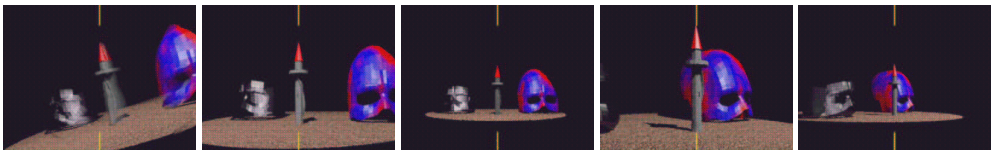


FIG. *4.2: Positionnement et suivi d'une trajectoire prédéfinie. L'animateur a spécifié une tâche de focalisation sur la tour (contraignant deux degrés de libertés). La caméra suit ensuite une trajectoire spécifiée par une spline 3D en utilisant une tâche secondaire (la distance entre le centre optique de la caméra et un point évoluant sur la courbe doit être minimale).*

Le problème de l'évitement d'obstacles (problème par ailleurs extrêmement classique qui a été largement étudié dans la littérature) peut se résumer de la façon suivante : une caméra mobile doit se focaliser (au sens d'une tâche visuelle définie par l'animateur) sur une cible d'intérêt (elle même éventuellement mobile) et ne doit pas entrer en collision avec les autres objets présents dans la scène (fixes ou mobiles). La solution sans doute la plus classique à ce problème est de considérer une approche de type champs de potentiel [95]. Drucker [59] utilise cette technique pour générer hors-ligne les mouvements de la caméra mais, dans ce système, les comportements des différents acteurs de l'environnement sont connus *a priori* avant de générer l'animation. Cette approche peut cependant s'appliquer en temps-réel de manière réactive en ne considérant que des potentiels locaux à la caméra. Une seconde solution, relativement proche de la première, consiste à utiliser les degrés de liberté restants à la caméra pour maximiser la distance caméra/obstacle [61]*[138]*.

Le problème de l'évitement d'occultations est différent : l'objectif est d'éviter qu'un objet s'interpose entre la caméra et l'objet d'intérêt ou cible. La solution classiquement utilisée pour résoudre ce problème (dans les jeux vidéo par exemple) est de maintenir le vecteur vitesse de la caméra aligné avec celui de la cible. Cette situation ne fonctionne en pratique que dans le cas où c'est le mouvement dû à la cible qui va provoquer l'occultation. Cette technique est en pratique très peu efficace. Comme nous l'avons exposé dans le chapitre précédent (section 3.2.2), les occultations peuvent être évitées en maximisant la distance dans l'image entre la projection de la cible et celle de l'objet occultant (voir la fonction de coût donnée par l'équation 3.2). Un critère reposant uniquement sur l'image n'est pas suffisant pour valider ou invalider le risque d'occultation. Dans le cas où cette tâche était réalisée dans un contexte robotique, il peut être difficile de définir avec certitude si un objet mobile va réellement occulter la cible. Dans le contexte de l'animation une connaissance totale de l'environnement à l'instant courant est disponible (incluant la position et la vitesse de tous les objets de la scène). Il est donc possible de prévoir et quantifier le risque d'occultation afin d'activer le processus d'évitement.

Dans l'exemple présenté sur la figure 4.3, nous avons appliqué cette méthodologie à une tâche de navigation dans un environnement complexe. La cible à suivre se déplace, avec

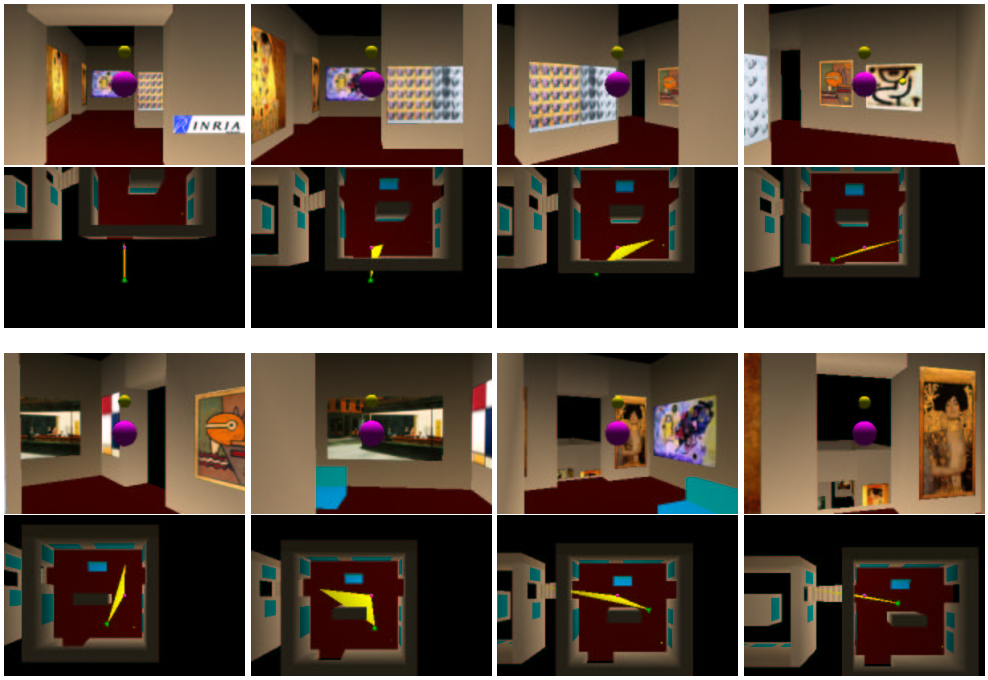


FIG. *4.3: Traversée d'un musée par une caméra poursuivant une cible. La tâche principale de la caméra est de rester focalisée sur sa cible (les deux sphères violette et jaune). Deux tâches secondaires sont par ailleurs considérées, spécifiant d'une part à la caméra de ne pas heurter un obstacle (ici les murs du musée) et d'autre part de toujours éviter l'occultation de la cible par des éléments du décor. Les première et troisième lignes montrent la vue de la caméra. Les secondes et quatrième lignes montrent une vue du dessus. Les volumes jaunes visibles sur ces vues du dessus sont utilisés pour prédire le risque d'occultation en extrapolant les positions futures de la cible. Les résultats de la même expérience dans le cas où aucune tâche secondaire n'est considérée sont présentées dans* [138].

un mouvement inconnu, dans un environnement de type musée. L'objectif est de maintenir la cible centrée dans l'image en évitant les obstacles et les occultations par les murs de la pièce tout en considérant *en ligne* les modifications de l'environnement (c'est-à-dire la présence d'autres objets mobiles, voir résultats dans [138]).

D'autres utilisations possibles de la redondance ont été proposées dans le cadre de l'animation d'humanoïde de synthèse et seront présentées dans la section 4.3.

4.2 Application : cinématographie virtuelle

La cinématographie virtuelle désigne la capacité de choisir les successions de points de vue depuis lesquels les environnements virtuels en 3D sont rendus et à calculer automatiquement les déplacements de la caméra pendant un plan (entre deux points de vue différents). Dans un contexte cinématographique, le choix des points de vues sert souvent la narration, de même, les mouvements de la caméra participent à créer une ambiance, à transmettre un contenu émotionnel. La succession de plans à tourner est classiquement

décrit par une suite de dessins (ressemblant fortement à une bande dessinée) : les *storyboards*. Pour chaque plan des annotations sont spécifiées, utiles au tournage (taille et durée du plan, angles de prise de vue, hauteur de la caméra, lumière, etc.), le contenu du cadre, les dialogues, des indications sur la musique et les bruitages, des informations sur les raccords. Toutes ces informations peuvent être directement encodées à haut niveau dans des automates (ou des structures de données équivalentes) et sous forme de tâches visuelles, permettant de contrôler finement les mouvement de la caméra, basées sur la technique d'animation [43] présentée dans la section précédente.

Fig. 4.4-a: Garde à vue *de Claude Miller*

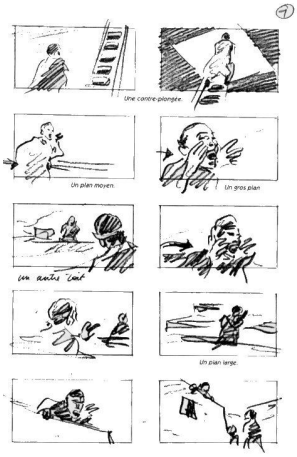


Fig. 4.4-a: Himalaya *de Claude Valli*

FIG. *4.4: Extraits de storyboards. Les plan à tourner ainsi que les différents mouvements de la caméra à réaliser sont spécifiés dans l'image. La synthèse de consignes visuelles pour l'asservissement visuel à partir d'une telle description semble donc naturelle.*

Les cinéastes ont établi de façon plus ou moins implicite un ensemble de règles et de conventions permettant de transmettre les informations d'une manière compréhensible et efficace [5]. Le niveau de description de ces règles et conventions n'est cependant pas assez détaillé pour définir une grammaire formelle du langage cinématographique et il existe certaines latitudes (nécessaires pour servir la dimension artistique) employées par les auteurs de film. Il reste que ces contraintes sont utilisables dans un contexte de cinématographie virtuelle (temps-réel ou non). Comme elles s'expriment le plus souvent dans l'image, il est aisé de les coder sous forme de tâches d'asservissement visuel. L'exemple le plus parlant est celui du dialogue entre deux personnes présenté sur la figure 4.5.

Dans un contexte plus large où les plans s'enchaînent, un système allouant automatiquement les ressources (caméras et tâches élémentaires) doit être construit. He [81] a développé un paradigme permettant le contrôle d'une caméra en temps réel appelé *the Virtual Cinematographer*, ce contrôle étant beaucoup plus proche du montage (au sens cinématographique du terme), que de la réalisation d'un plan. Le système créé repose sur l'existence d'un certain nombre de plans typiques ou idiomes organisés de manière hiérarchique, et génère une succession de ces plans avec un *timing* particulier dans le but de

FIG. *4.5: Cinématographie virtuelle : exemple de gestion de contraintes cinématographiques pour filmer un dialogue entre deux personnages. Le schéma de gauche illustre le principe du champ contre-champ interne ou externe [5] : "L'approche la plus simple pour un dialogue face à face est l'utilisation d'un système d'angles opposés externes. Quand l'acteur apparaît en premier plan (de dos vers nous), en contre-champ externe, le bout de son nez ne devrait pas dépasser la ligne de sa joue [...] La répartition de l'espace scénique en un-tiers deux-tiers est fondamentale, bien que des variantes puissent être utilisées si on le désire.". Les deux images gauche montrent deux plans se conformant à ces spécifications ; les objectifs visuels sont en jaune dans l'image. Ces images sont issues d'une démonstration présentée au festival Imagina 2002 à Monte Carlo)*

traduire au mieux l'ensemble des événements se produisant. L'approche qui a été retenue dans [43]*[44]* est similaire à celle de He [81]. Les règles cinématographiques sont encodées dans des automates à états finis. Chaque état de l'automate correspond à une prise de vue ("*shot*") particulière.

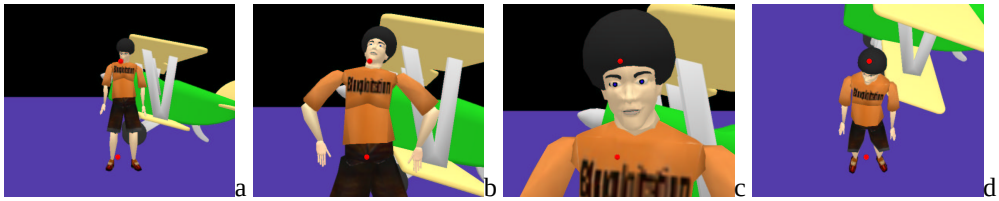


FIG. *4.6: Enchaînement entre différents plans. Les points rouges montrent les objectifs visuels dans l'image (a) plan d'ensemble (b) plein cadre (c) gros plan (d) un autre plan d'ensemble avec contre-plongée*

Pour chaque prise de vue, contrainte par une règle cinématographique élémentaire, un "module" gère les positionnements de la caméra pour différents types de prise de vue (dialogue, traveling, gros plan, plan américain, plongée, contre plongée, etc). Ces "modules" élémentaires sont bien évidemment constitués de tâches d'asservissement visuel. Les transitions entre états correspondent à des changements de prise de vue qui peuvent être des coupures franches (on passe d'une caméra à une autre), ou des transitions animées (la même caméra virtuelle est utilisée mais la tâche visuelle est modifiée, voir figure 4.6).

Un autre problème majeur en cinématographie, mais peu abordé dans le cadre de la cinématographie virtuelle, est celui de la la photographie. Maîtriser la lumière permet de contrôler en permanence le rendu artistique de la scène ou du sujet à filmer [150]. La première difficulté réside dans la détermination des bons critères permettant "d'optimiser" l'éclairage. Deux critères sont donc proposés pour parvenir à cet objectif. Le premier repose sur une maximisation de la quantité de lumière réémise par l'objet et le second repose sur les

gradients d'intensité de l'image (qui donne une information sur le contraste). L'objectif fondamental étant de mettre en valeur les caractéristiques principales qui illustrent le sujet, l'éclairage doit souligner les formes et les mettre en valeur. De fait le second critère reposant sur les contrastes est beaucoup plus adapté. Pour attaquer ce problème, notre objectif initial a été de déterminer la position de la caméra assurant au mieux ces contraintes. Dans un deuxième temps, afin de maintenir inchangé l'aspect de l'objet d'intérêt, nous avons cherché uniquement à modifier la position de la source lumineuse *[138]*. Ces études ont été validées sur de nombreux exemples (voir par exemple l'illumination de la Vénus de Milo sur la figure 4.7).

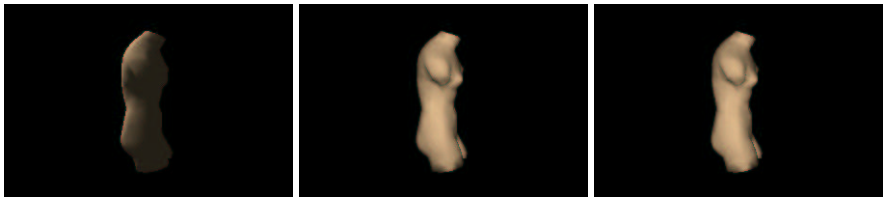


FIG. *4.7: Illumination de la Vénus de Milo : ici la tâche vise à maximiser le contraste dans l'image*

4.3 Commande d'un humanoïde de synthèse

Les sections précédentes étaient dédiées à l'animation d'une caméra dans un environnement virtuel. Dans le formalisme retenu, la commande envoyée à la caméra dépend de ce que "perçoit" celle-ci. L'analogie avec un humanoïde virtuel est directe ; la plupart des actions entreprises par celui-ci sur son environnement extérieur dépendent de la perception qu'il en a. L'un de nos objectifs a donc été d'étudier la manière dont on peut adapter le formalisme précédemment décrit au problème du contrôle d'un humanoïde virtuel, en modélisant la cinématique associée à son attention visuelle dans un contexte de perception active.

4.3.1 Techniques d'animation d'un humanoïde

L'animation d'humanoïde se trouve au carrefour de plusieurs domaines applicatifs (mondes virtuels, films de synthèse, jeux vidéos, recherche biomécanique, etc.). La principale exigence que l'on peut avoir vis-à-vis de l'animation d'un humanoïde se situe au niveau du réalisme de son comportement tant au niveau de la cohérence des mouvements avec les modèles physiques et biomécaniques (ceci découle directement de la technique d'animation retenue) que de la cohérence avec des modèles comportementaux (c'est-à-dire sur la cohérence globale des actions de l'humanoïde).

Le problème principal dans le cadre de l'animation d'un humanoïde revient à générer un mouvement réaliste, au travers de la gestion de trois paramètres : position, orientation et déformation de la géométrie sous-jacente). Les techniques utilisées pour l'animation d'un humanoïde sont donc issues des techniques d'animation générale de solides rigides

ou déformables [82]. Depuis l'utilisation de l'animation image par image, de nombreuses techniques d'animation ont émergé, pouvant se regrouper en trois catégories :

- les méthodes dites *cinématiques*, qui cherchent à reproduire un certain nombre d'effets sans s'intéresser aux causes, à partir d'une description de ce que doit être le mouvement. Dans le cas de la cinématique directe le problème consiste à décrire la trajectoire angulaire de chaque articulation du squelette de l'humanoïde. Dans le but de contrôler la structure, on cherche souvent à résoudre le problème inverse [192, 11], c'est-à-dire retrouver les paramètres des valeurs angulaires de chaque articulation à partir des positions finales des effecteurs. Ce problème est particulièrement étudié en robotique dans le cadre des rigides articulés (bras mécaniques, bipèdes). La qualité des mouvements obtenus dépend des spécifications des positions finales des effecteurs. Par ailleurs cette méthode peut s'avérer coûteuse en temps de calcul. Une comparaison entre les techniques de cinématique directe et inverse est proposée dans [19].
- la génération du mouvement à partir de mouvements, traditionnellement acquis grâce à des systèmes de *motion capture*. Ces techniques sont apparues dans le courant des années 90 et consistent à enregistrer le mouvement d'une partie du corps d'un humain à l'aide d'un système de capture de mouvement pour l'intégrer ensuite à un modèle virtuel. La difficulté principale soulevée par ce type de techniques consiste à générer de nouveaux mouvements sur la base de mouvements existants. Plusieurs méthodologies existent pour résoudre ce problème. Citons par exemple la composition linéaire des mouvements, ou *motion blending* [206, 168] ou encore la modification du mouvement enregistré ou *motion warping* [112, 164].
- les méthodes dites *dynamiques*, dont les fondements sont les causes du mouvement. Les lois de la physique (donc de la mécanique) se substituent alors aux connaissances a priori sur le mouvement. L'avantage des méthodes dynamiques est de pouvoir générer des mouvements très réalistes. Cependant, les équations différentielles décrivant les lois mécaniques sont généralement fortement non-linéaires et présentent des termes de couplage rendant difficiles leur résolution [209]. Cette méthode est donc difficile à employer pour un contrôle au niveau tâche de l'humanoïde.

Les techniques d'animation du squelette de l'humanoïde sont donc nombreuses, et le choix de la technique dépend finalement beaucoup du cadre applicatif. La tendance actuelle est à l'utilisation, dans le cadre du contrôle d'un humanoïde, de modèles mixtes, c'est à dire utilisant les différentes techniques présentées. Cette mixité des "opérateurs" d'animation s'exerce soit dans le cadre de modèles de mouvements précis, par exemple, mélange de mouvements capturés et de cinématique inverse [186] ou association de la cinématique avec la distribution des masses (cinématique inverse) [19], soit à l'intérieur d'une même architecture visant à réunir toutes les possibilités de contrôle pour un humanoïde, avec éventuellement des contrôles distincts pour les différentes parties du corps. Ainsi dans le laboratoire de recherche sur les modèles d'humanoïde de l'université de Pennsylvanie, le modèle d'humanoïde *Jack* [9] dispose de plusieurs modes de contrôle. Il en est de même pour les humanoïdes développés au VRLab de l'EPFL ainsi qu'au MiraLab [120].

Les techniques cinématiques semblent les plus intéressantes dans le cadre du contrôle temps-réel d'un humanoïde virtuel, notamment grâce à la flexibilité de leurs modes opératoires. Cependant les animation résultantes peuvent parfois manquer de réalisme et la spécification de la tâche peut s'avérer complexe. Dans l'idéal, la simulation de la percep-

tion d'un humanoïde virtuel requiert l'utilisation d'une méthode permettant :

- de spécifier des tâches simples, la simplicité pouvant venir d'une description proche du langage naturel, ou se traduisant par des paramètres de contrôle simples,
- d'être temps-réel et robuste aux modifications de l'environnement, et de même être capable de changer de type de tâche aisément,
- de proposer un formalisme flexible permettant d'introduire des contraintes dans les mouvements générés.

L'utilisation d'une technique référencée vision permet de combler en partie ces lacunes, comme nous allons le montrer dans la partie suivante.

4.3.2 Animation à l'aide de l'asservissement visuel

Nous nous sommes donc intéressés à la modélisation de l'animation associée à l'attention visuelle d'un humanoïde à l'aide de l'asservissement visuel. L'analogie observée entre les problèmes de contrôle de caméra et du contrôle de l'humanoïde amène à envisager l'utilisation du même formalisme. L'idée est de pouvoir lier une chaîne articulaire représentant une partie du corps de l'humanoïde impliquée dans le processus d'attention visuelle à la perception de ce qu'il "voit" au travers de ses "yeux", représentant l'effecteur final de la chaîne.

La simulation de l'attention visuelle d'un humanoïde de synthèse met en jeu différentes parties du corps humain. De prime abord, ce sont avant tout les yeux qui jouent le rôle le plus important : ce sont eux qui orientent le regard. Le cou, le torse et le reste du tronc (c'est-à-dire l'ensemble de la colonne vertébrale) font aussi partie de ce processus. Pour représenter cette chaîne, la paramétrisation classique de Denavit-Hartenberg a été utilisée. Elle offre une représentation unifiée et permet d'inclure des informations telles que le poids des différentes parties ou la position des centres de masse, ce qui peut être utile dans ce contexte. Une fois cette modélisation réalisée, l'objectif est de pouvoir contrôler la cinématique de la chaîne articulaire impliquée dans le processus d'attention visuelle. L'approche retenue consiste donc à considérer l'humanoïde (ou plus exactement la chaîne articulaire sous-jacente) comme un robot. Les yeux de l'humanoïde étant alors assimilés à une caméra montée sur l'effecteur d'un robot. L'utilisation du Jacobien de ce "robot" permet d'exprimer directement la commande calculée dans l'espace articulaire *[32]* (voir équation 2.3). Cette approche est donc extrêmement liée aux techniques de cinématique inverse. À chaque pas de temps une nouvelle commande en vitesse est donc envoyée à la chaîne articulaire. Cette commande en vitesse se traduit après intégration en une nouvelle configuration de la chaîne articulaire.

Nous avons utilisé une chaîne articulaire composée de neuf degrés de liberté. Ces neuf degrés de liberté sont des liaisons de type rotoïde. Ils représentent respectivement l'abdomen (trois rotations), le thorax (un seul degré), le cou (trois rotations) et les yeux (deux rotations). À l'aide de cette chaîne, nous avons réalisé différentes tâches de positionnement [43]. Le comportement associé à ce genre de tâches peut être décrit comme un comportement d'attention visuelle, l'humanoïde cherche à regarder un point précis de la scène. Dans l'exemple présenté dans la figure 4.8, l'humanoïde doit observer une balle. L'information visuelle retenue est donc une sphère, ce qui implique l'introduction d'une contrainte sur la distance finale entre les yeux de l'avatar et la balle. On constate [43] que toutes les ar-

ticulations sont impliquées dans le mouvement même si uniquement trois degrés de liberté sont contraints par la tâche.

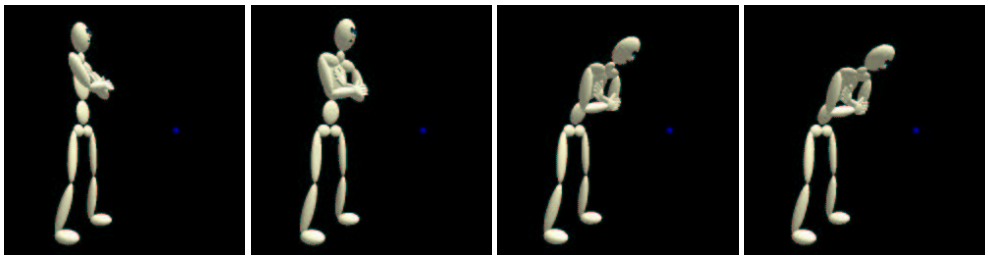


FIG. *4.8: Tâche de positionnement d'un humanoïde de synthèse par rapport à une balle.*

Plusieurs raisons justifient le choix de cette technique pour l'animation d'un humanoïde : simplification des paramètres d'entrée, mimétisme avec la perception humaine dans la spécification de ces entrées, adaptation dynamique et temps réel du système dans des environnements inconnus (grâce à une boucle fermée intégrant des retours du modèle réel). Le principal défaut de cette approche est d'une certaine manière identique à celui rencontré dans le cadre du contrôle d'une caméra : on ne dispose que d'un contrôle lâche sur les trajectoires articulaires, pouvant conduire à des mouvements non-réalistes. La gestion de ce réalisme peut cependant se faire en introduisant des contraintes dans les mouvements générés. Deux contraintes ont principalement été considérées : l'évitement des butées articulaires et la tendance à se rapprocher d'une posture de moindre effort [19]*[43]*. Il est absolument indispensable de considérer la première contrainte gérant les butées articulaires. Dans le cas d'un humanoïde, la position de ces butées est bien connues (en moyenne, chaque individu disposant de ses propres caractéristiques anatomiques) et l'intégration de cette contrainte se fait en utilisant l'approche présentée dans la section 3.2.1 (voir *[32][46]* pour plus de détails). Contraindre l'humanoïde à se retrouver dans une position "mécaniquement" acceptable ne garantit pas le réalisme du mouvement. La loi de commande utilise en effet tel ou tel degré de liberté pour assurer la contrainte imposée, alors qu'un opérateur humain opérera différemment et choisira (consciemment ou inconsciemment) le geste de moindre effort. L'idée est donc d'avoir une fonction secondaire [19] agissant comme une force de rappel vers cette position de repos (sous la contrainte que la tâche visuelle soit maintenue). Les résultats présentés sur la figure 4.9-a montrent l'intérêt de cette approche dans le cas d'une tâche de poursuite. On constate par ailleurs que le comportement résultant de l'humanoïde est conforme au modèle physiologique de perception [167] (figure 4.9-b). Afin d'augmenter encore le réalisme, d'autres contraintes (comme celle portant sur la distribution des masses dans le corps [19]) devraient être introduites.

Si l'on considère l'attention visuelle au niveau tâche, il apparaît que la simple cinématique du buste, de la tête et des yeux de l'humanoïde ne suffit pas. Plusieurs mécanismes sont mis en jeu, dont notamment la locomotion, ou d'autres gestes comme s'accroupir, etc. Ce problème est extrêmement difficile et nous n'y avons que peu contribué. La solution retenue a été de simplement rajouter deux degrés de translation à la base de la chaîne articulaire Ces translations correspondent à la marche de côté et à la marche normale (avancer et reculer). A chaque pas de temps, ces vitesses sont recalculées, et interprétées par un

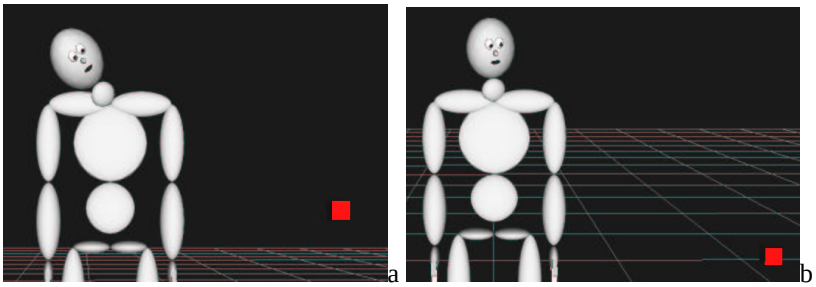


Fig. 4.9-a: *Tâche de suivi intégrant ou non une stabilisation vers une position de moindre effort. L'humanoïde doit se focaliser sur un objet mobile. L'image de droite représente la posture de l'avatar après 3000 itérations si uniquement une contrainte sur les butées articulaires est considérée. Dans l'image de gauche (acquise au même moment), la tâche intègre en plus une contrainte visant à maintenir l'avatar dans une position de repos. Dans les deux cas la tâche visuelle est réalisée et la position est mécaniquement admissible, cependant seule la seconde position est réaliste.*

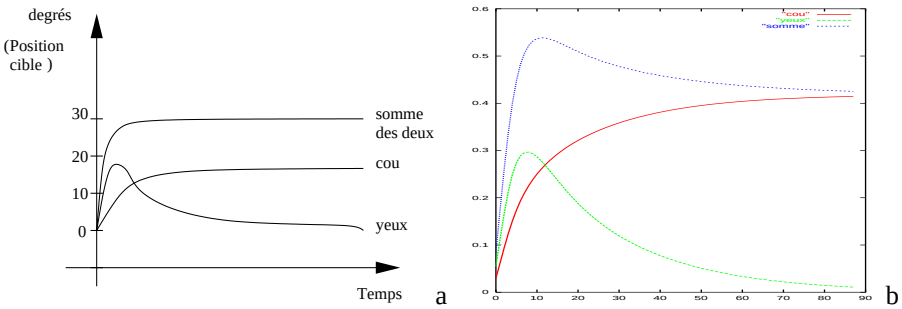


Fig. 4.9-b: *Coopération entre le mouvement du cou et le mouvement des yeux pour le suivi visuel d'une cible (réflexe vestibulo-oculaire). Comparaison entre l'étude de Robinson [167] (courbe de gauche) et les résultats obtenus avec notre système d'animation (courbe de droite)*

FIG. *4.9: Introduction de contraintes visant à améliorer le réalisme de l'animation : gestion de butée articulaire de l'humanoïde et stabilisation sur une position de repos*

module de locomotion décrit dans [147]. Cette action peut gérer plusieurs types de marche (avant, arrière et sur le côté) issus de mouvements capturés. Nous avons étudié la validité de cette approche au travers de l'exemple suivant : la tâche de vision donnée à l'humanoïde est de maintenir une certaine distance entre ses yeux et une balle qui se déplace. Le mouvement de cette balle est perpendiculaire à la direction de l'humanoïde. La primitive sur laquelle on s'asservit est un point situé au centre de la balle. Une seconde contrainte sur la distance relative yeux-balle est aussi spécifiée. Lors de la première phase de l'animation (figure 4.10, première ligne), l'humanoïde doit avancer vers la balle pour assurer la contrainte de distance, ainsi que jouer sur les rotations associées au cou et aux yeux pour la tâche d'observation. Dans un deuxième temps, l'humanoïde adopte une marche en pas

chassé pour compenser les mouvements de la balle. La transition entre la marche avant et les pas de coté s'est effectuée de manière fluide au moment où l'humanoïde arrive près de la balle. Une tâche similaire est montrée sur la figure 4.11 où l'humanoïde est asservi sur une sphère.

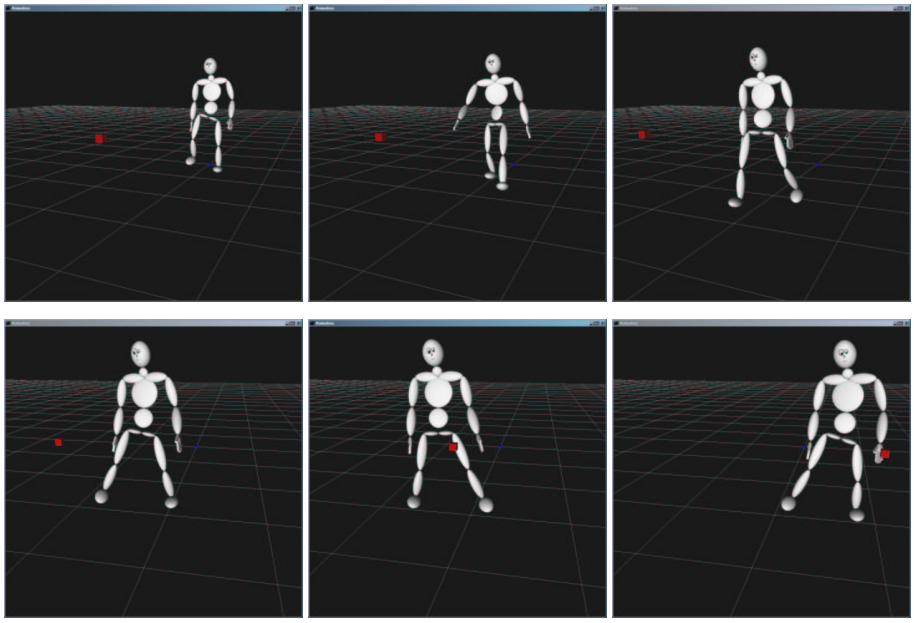


FIG. *4.10: Gestion de la locomotion. L'humanoïde doit regarder de près un objet qui bouge : (première ligne) l'humanoïde se rapproche de la balle (seconde ligne) puis effectue des pas de coté pour maintenir la tache visuelle ainsi que la distance souhaitée.*

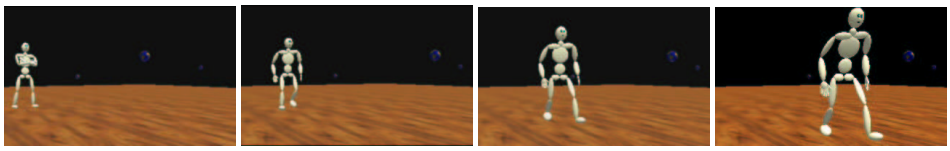


FIG. *4.11: Exemple de gestion de la locomotion avec une chaîne articulaire comportant douze degrés de liberté (dont trois en translation) : asservissement sur une sphère*

En marge des travaux réalisés sur le contrôle des humanoïdes de synthèse par asservissement visuel, nous avons mené quelques travaux préliminaires sur la simulation de la perception visuelle qui semblent constituer une piste de recherche intéressante pour le choix automatique des éléments visuels à observer *[47]*. Dans le cadre de l'animation l'objectif est de donner plus de réalisme aux scènes produites en sélectionnant pour l'humanoïde automatiquement et selon des critères psychovisuels les points de fixation. Les informations saillantes sont extraites en utilisant d'une part des filtres de Gabor qui modélisent relativement bien le comportement de la rétine [125] et d'autre part un estimateur de mou-

vement robuste utilisé pour la détection des zones saillantes au sens du mouvement. La figure 4.12 montre les résultats obtenus sur une image. Ces travaux initiés dans un contexte de réalité virtuelle ont été aussi validés et étendus en robotique dans un contexte de télésurveillance *[45]*.

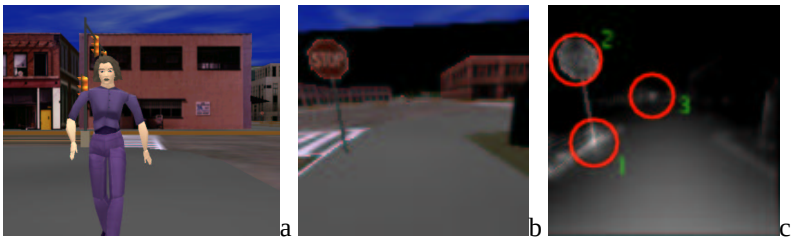


FIG. *4.12: Utilisation de carte de saillance en environnement urbain. Les mouvements de locomotion de l'humanoïde sont générés de manière arbitraire, par contre, les mouvements de la tête et des yeux sont générés automatiquement par asservissement visuel afin de se focaliser sur les zones les plus saillantes. L'image (a) montre l'humanoïde se déplaçant dans son environnement, l'image (b) montre sa perception de l'environnement et finalement, l'image (c) montre le résultat du calcul des cartes de saillance avec les trois points les plus saillants.*

Bilan

Dans ce travail nous avons abordé des problèmes de contrôle relatifs à l'animation et à la simulation en synthèse d'images. Plus spécifiquement, nous nous sommes intéressés aux liens entre la perception et l'action au travers de deux grandes classes de problème : l'animation d'une caméra virtuelle dans un environnement synthétique et la simulation d'un humanoïde de synthèse. Nous avons défini un cadre original en animation : l'animation référencée vision. Cette technique permet d'obtenir des mouvements 3D d'une entité géométrique à partir d'une consigne spécifiée dans un espace 2D. Un des intérêts majeurs de cette technique est de permettre une simplification des paramètres de contrôle (économie de spécifications), ainsi qu'une manipulation plus aisée de ces paramètres car plus proche des spécifications issues du langage naturel. Les différents résultats ont prouvé que cette approche est flexible, et peut s'adapter à un grand nombre de problèmes.

Il faut cependant reconnaître que cette méthode n'est pas exempte de défauts. Le principal défaut est la contrepartie directe de sa principale qualité : le contrôle se faisant dans l'image, on perd le contrôle de la trajectoire 3D de la caméra. Cette trajectoire 3D est calculée automatiquement pour assurer la tâche visuelle et les tâches secondaires mais l'animateur ne la maîtrise pas. Ce problème pourrait être en partie résolu en considérant dans la commande un mélange d'information 2D et 3D.

CHAPITRE 5

Suivi d'objets : vers un asservissement visuel virtuel

La définition d'algorithmes de suivi d'objets dans des séquences d'images a aussi été un de nos axes de recherche important. Un processus fiable d'extraction puis de suivi spatio-temporel de l'information visuelle est en effet une des clés du succès, ou de l'échec, d'une tâche d'asservissement visuel. Il apparaît d'autre part primordial pour introduire les techniques d'asservissement visuel dans un large éventail d'applications, de pouvoir appréhender des scènes naturelles, c'est-à-dire, sans marqueurs, avec des objets polyédriques ou non, et des conditions d'illumination variables, ... Historiquement, l'utilisation de marqueurs (voir figure 5.1a) a permis de valider les techniques d'asservissement visuel. Si le suivi de marqueurs reste la solution privilégiée pour la validation des lois de commande (voir le chapitre 3), s'y limiter serait un frein au transfert de ces technologies dans le monde industriel.

Suivi de primitives 2D. Cette approche consiste à décrire l'objet à suivre à l'aide de primitives géométriques comme des points particuliers [119, 177], des angles, des contours [14, 15], des segments de droites [75, 18]*[126]* ou des ellipses [199] *[126]*, etc. Cette approche est la plus classiquement utilisée dans un contexte d'asservissement visuel car il existe souvent une relation directe entre les primitives 2D suivies dans l'image et les primitives utilisées dans la loi de commande.

Dans ce contexte, le système XVision [75] est un bon exemple de ce type d'approche. Des primitives simples (droites, contours) sont suivies en temps-réel ; chaque point du contour est mis en correspondance dans l'image suivante par une simple recherche le long de la normale au contour. Le suivi d'objets plus complexes est possible en combinant plusieurs primitives élémentaires liées ensemble par un jeu de contraintes. Dans ViSP (la plateforme d'asservissement visuel que nous avons développée *[126]*), une telle fonctionnalité a aussi été proposée. Le suivi spatio-temporel de primitives élémentaires (droites, ellipses, courbes) repose à la base sur l'appariement d'éléments de contours en utilisant l'approche

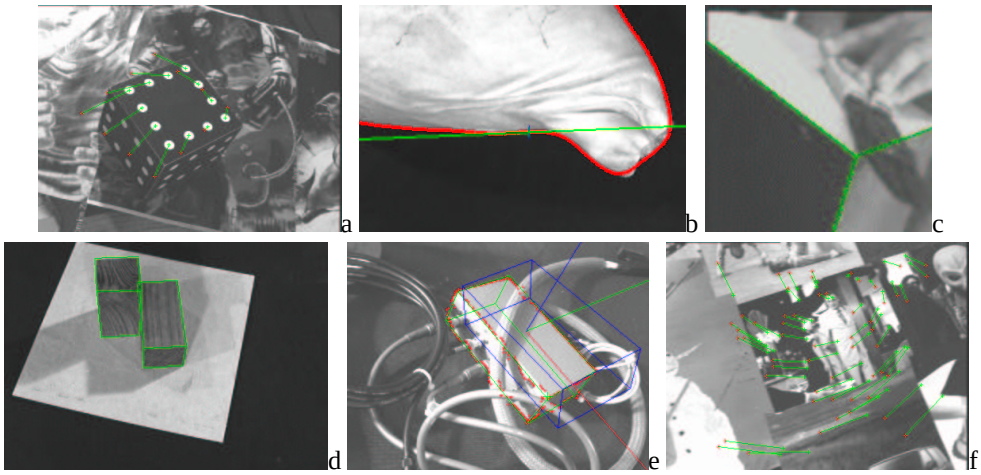


FIG. *5.1: Quelques exemples de scènes considérées dans des expérimentations d'asservissement visuel (classé par ordre – subjectif – de complexité croissante)*

dite des éléments de contour en mouvement ou ECM [20]. L'avantage principal de ces approches réside dans leur simplicité de mise en œuvre, et leur rapidité. Par contre, elles ne permettent pas, ou difficilement, le suivi de motifs complexes qui ne peuvent pas être modélisés à l'aide de primitives locales ou de contraintes uniquement 2D. De plus, la qualité des résultats dépend fortement du contenu de la scène et est très sensible à la densité des primitives dans l'image ainsi qu'aux phénomènes d'occultation.

Suivi de régions. Les méthodes précédentes reposent, de différentes manières, sur l'analyse des gradients de l'image. Une autre possibilité est de considérer directement l'intensité lumineuse et de faire un recalage spatio-temporel d'une partie entière de l'image sans extraire préalablement de primitives particulières. Le processus de recalage consistera alors à rechercher un jeu de paramètres décrivant la transformation ou le déplacement de la zone considérée entre les deux images en minimisant un certain critère de corrélation (e.g., [84]). L'adoption d'une recherche exhaustive des transformations minimisant ce critère de corrélation est très peu performante et il est possible de résoudre le problème via un processus de minimisation, linéaire ou non, permettant de prendre en compte des mouvements complexes (affines, homographiques, etc.). Une méthode récente, que l'on peut considérer comme une approche différentielle, permettant de considérer les variations des paramètres de mouvement comme une fonction linéaire d'une image de différence [75, 74] n'est pas sans lien avec la formulation classique de l'asservissement visuel (une matrice d'interaction relie en effet la variation de l'intensité lumineuse à la variation des paramètres de mouvement). Une extension de cette approche permettant un apprentissage de cette matrice est présentée dans [91]. Ce nouvel algorithme permet d'augmenter la taille du cône de convergence et est donc plus robuste. Il est a noter que ces travaux sont par ailleurs très proches de méthodes proposées pour l'estimation du mouvement dominant [159].

Suivi basé modèle. Afin de pouvoir prendre en compte des mouvements quelconques de l'objet tout en introduisant des contraintes très fortes dans le processus de mise en correspondance spatio-temporelle, il peut être intéressant de considérer un modèle de l'objet. Dans le cas où ce modèle est un modèle 2D des objets ou des formes que l'on cherche à suivre, il est nécessaire d'introduire une composante déformable pour pouvoir s'adapter aux déformations non linéaires des projections de l'objet dans le plan image liées à des effets de perspective non pris en compte par ces modèles 2D [90]. Il existe un large éventail de modèles déformables plus ou moins complexes allant des modèles à structure peu contrainte, comme les modèles de contours actifs [14, 92, 15], aux approches par forme prototype – les "templates déformables" – [152, 41, 94]. Il est par ailleurs possible de considérer des modèles 3D, généralement un modèle CAO de l'objet, le problème revient alors à déterminer l'attitude de l'objet à l'aide d'une vue [58, 117, 70, 98, 52, 194, 60, 198]*[36]*, ou de plusieurs [22, 142]. Ces derniers cas permettent d'obtenir une meilleure précision et de traiter des occultations temporaires de l'objet dans une des vues. Ce type d'approche reposant sur des modèles 3D se prête très bien à une intégration dans des expérimentations d'asservissement visuel que ce soit au sein de loi de commande 2D, 2D 1/2 ou 3D [102, 194].

Nous nous sommes donc intéressés au développement d'algorithmes de suivi de primitives ou d'objets dans des séquences d'images. Ce point est d'autant plus difficile à traiter dans le contexte de l'asservissement visuel qu'il faut en même temps satisfaire des contraintes de temps-réel (ou proches du temps-réel) et de précision pour que la méthode de suivi puisse être exploitable. Certains des algorithmes que nous avons développés ont par ailleurs trouvé une applications dans le contexte de la réalité augmentée.

5.1 Suivi 2D de formes simples pour l'asservissement visuel.

Afin de pouvoir considérer le suivi de primitives géométriques *a priori* quelconques (e.g., droite, ellipse, courbe spline, etc.), il est nécessaire de se donner un cadre générique permettant de suivre localement des points de contour, puis *a posteriori* de calculer de façon robuste les paramètres de la primitive considérée (par des technique de type moindres carrés pondérés). Pour satisfaire ces contraintes de robustesse et de rapidité, nous avons exploité l'algorithme dit des ECM (Éléments de Contours en Mouvement) [20]. L'avantage de cette méthode est qu'elle ne nécessite pas d'étape, souvent coûteuse, d'extraction des contours spatiaux dans l'image et n'est ainsi pas sujette aux aléas de segmentation. De plus, elle peut être implantée en temps réel, les calculs à effectuer se limitant à de simples convolutions par des masques pré-calculés [18, 20].

Le traitement consiste à calculer, pour chaque point du contour considéré, la composante de déplacement perpendiculaire au contour entre deux images à t et $t + \Delta t$. Elle ne nécessite que l'application d'une convolution en chaque position candidate, et est très performante d'un point de vue du temps de calcul. Le procédé consiste à rechercher le "correspondant" $\mathbf{p}^{t+1}$ dans l'image I^{t+1} de chaque pixel $\mathbf{p}^t$ (voir Figure 5.2). Nous déterminons un intervalle de recherche 1D $\left\{\mathbf{q}^j, j \in [-J, J]\right\}$ dans la direction δ normale à l'arête projetée. Pour chaque position $\mathbf{q}^j$ dans l'image I^{t+1} dans la direction δ, nous calculons le

critère de mise en correspondance qui s'exprime comme la maximisation d'un rapport de vraisemblance ζ^j. Ce rapport peut se réécrire comme la valeur absolue de la somme des convolutions dans l'image calculée en $\mathbf{p}$ et $\mathbf{q}^j$ avec un masque pré-calculé $\mathbf{M}_\delta$ dépendant de l'orientation de l'arête projetée. La nouvelle position $\mathbf{p}^{t+1}$ est donnée par [20] :

$$\mathbf{p}^{t+1} = \arg \max_{j \in [-J,J]} \zeta^j \text{ avec } \quad \zeta^j = \mid I^t_{\nu(\mathbf{p})} * \mathbf{M}_\delta + I^{t+1}_{\nu(\mathbf{q}^j)} * \mathbf{M}_\delta \mid$$

où $\nu(.)$ désigne un voisinage du pixel considéré. Cette formulation est beaucoup plus robuste qu'une simple recherche du maximum de gradient le long de la normale au contour en utilisant des masques dérivateurs classiques. D'une part, les masques sont orientés en fonction de la normale au contour et d'autre part en utilisant cette formulation, la réponse de la convolution en $\mathbf{p}^t$ et en $\mathbf{p}^{t+1}$ est implicitement comparée ce qui implique donc que les deux points homologues doivent se ressembler. Ceci introduit une contrainte spatio-temporelle forte qui peut cependant s'avérer préjudiciable dans le cas où l'objet à suivre est situé sur un fond texturé. Une fois l'ensemble des points de contour appariés, il suffit de déterminer les paramètres de la primitive considérée par un ajustement aux moindres carrés. Des techniques robustes [86, 169] peuvent bien sur être utilisées pour cela.

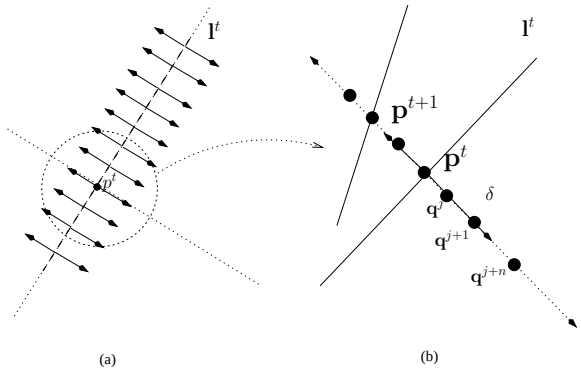


FIG. *5.2: Principe de l'algorithme des ECM pour la mise en correspondance de points de contour (a) calcul de la normale le long d'un contour, (b) recherche du correspondant le long de la normale.*

La figure 5.3 montre le résultat du suivi d'un certain nombre de primitives visuelles lors de leur utilisation au sein d'expérimentation en asservissement visuel.

5.2 Suivi 3D : localisation d'une caméra.

La méthode de suivi 2D évoquée dans le paragraphe précédent peut donner des résultat satisfaisants dans un environnement relativement maîtrisé où les contours des objets sont bien marqués. Elle permet par ailleurs un suivi rapide totalement compatible avec les cadences imposées par l'asservissement visuel. En présence de bruit, de fonds texturés, ou d'autres perturbations, les contraintes introduites dans la structure géométrique de la forme suivie ne sont cependant plus suffisantes pour permettre un suivi robuste. Dans tous

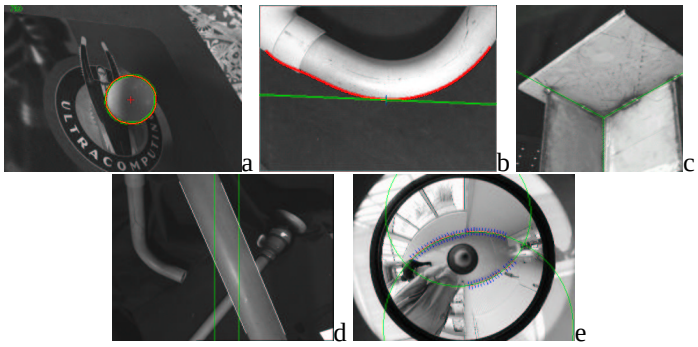


FIG. *5.3: Suivi de primitives 2D simples reposant sur la méthode des ECM dans des expérimentations utilisant l'asservissement visuel (a) reconstruction d'une sphère par vision dynamique active* [31]*, (b) suivi de contour [126], (c) suivi d'un trièdre* [4]*, (d) positionnement par rapport à un cylindre avec évitement des butées* [32]*, (e) suivi de "droites" prenant la forme de cercles dans des images catadioptriques (projet* ROBEA OMNIBOT [149]*)*

les autres travaux que nous avons réalisés sur le suivi d'objets, nous avons considéré une contrainte 3D, à savoir des connaissances sur la structure 3D de l'objet. Cette contrainte très forte permet d'effectuer un recalage 2D-3D qui rend le suivi beaucoup plus robuste tout en permettant une localisation de l'objet suivi par rapport à la caméra.

Le suivi se transforme alors en un problème de calcul de pose ou de localisation 3D. Si, de plus, les paramètres internes de la caméra ne sont pas disponibles, on a affaire à une problème d'étalonnage de la caméra. Ce sont deux problèmes très anciens en vision par ordinateur (citons les travaux sur le P4P [65] – *Perspective from 4 points* –) et en photogrammétrie [23] mais qui ont suscité et continuent de susciter de nombreuses études. Différentes approches ont été développées pour estimer la position relative de la caméra par rapport à la scène et il nous semble impossible de les énumérer ici de façon exhaustive.

Considérons, pour illustrer ce point, le problème de la localisation 3D à partir de la projection de points. Ce problème de recalage 2D-3D revient à déterminer les paramètres extrinsèques de la caméra, définis par la matrice homogène de changement de repère ${}^{c}\mathbf{M}_{o}$, qui minimise l'erreur de reprojection suivante :

$$\Delta = \sum_{i=1}^{N} \left(\mathbf{p}_i - \mathbf{K}\,{}^{c}\mathbf{M}_{o}\,{}^{\mathbf{o}}\mathbf{P}_i\right)^2 \tag{5.1}$$

où ${}^{\mathbf{o}}\mathbf{P}$ représente la position des N points considérés dans un repère lié à la scène (modèle de l'objet), $\mathbf{p}_i$ leur projection dans le plan image et $\mathbf{K}$ est la matrice de projection perspective. On peut cependant établir deux classements transversaux : le premier porte sur la méthode retenue pour réaliser le recalage entre les primitives 2D extraites des images et leurs correspondants 3D, le second porte plus sur la nature des informations visuelles extraites des images et utilisées lors du recalage.

Le premier facteur discriminant porte donc sur la méthode retenue pour réaliser le recalage 2D-3D. Dans le cas où un faible nombre de primitives est disponible, il existe des solutions purement analytiques à ce problème consistant à résoudre directement le système

d'équations non-linéaires issu de l'équation (5.1). Pour 3 points, il y a 4 solutions à ce problème [65], mais des solutions uniques existent pour un nombre de 4 points [83] ou de 4 droites [58]. Par nature, ces problèmes sont non linéaires par rapport aux paramètres de pose mais il existe des solutions linéaires (e.g., [67, 64, 115]) reposant sur la résolution de systèmes linéaires aux moindres carrés pour estimer la pose et éventuellement les paramètres intrinsèques de la caméra. Dans ce cas, l'efficacité de ces approches repose principalement sur la représentation choisie pour la matrice de rotation (matrice 3×3 [67, 64], angles d'Euler [115]) et des contraintes retenues pour assurer l'orthonormalité de cette matrice. Elles sont cependant extrêmement sensibles aux bruits de mesure.

Ces méthodes ne fournissant généralement pas un résultat de très bonne qualité, d'autres approches comme les techniques de minimisation non-linéaire (e.g., [116, 117, 60, 142, 52]*[134]* pour le calcul de pose ou [23, 33] pour l'étalonnage) peuvent alors être considérées. Elles consistent à minimiser l'erreur entre les observations dans l'image et la projection du modèle de l'objet pour une pose donnée. La minimisation est généralement réalisée en utilisant des algorithmes numériques itératifs de type Newton-Raphson ou Levenberg-Marquardt. Le principal avantage de ces approches est la précision du résultat obtenu. En contrepartie, l'algorithme de minimisation est sensible aux minima locaux, et peut, dans certains cas critiques, diverger. C'est pourquoi, une bonne initialisation du vecteur de paramètres à estimer est souvent nécessaire. Pour pallier ces difficultés, des techniques multi-étapes (e.g., [196, 203, 202]), plus dédiées à la calibration, considèrent une estimation linéaire (en ajoutant des contraintes comme la contrainte de Tsai sur l'alignement radial pour linéariser le problème) de certains paramètres et réalisent une estimation itérative des autres. Ces algorithmes autorisent une convergence plus rapide vers une bonne solution.

Finalement une solution très élégante est proposée dans [56, 158]. Cette approche repose sur un algorithme itératif faisant initialement l'hypothèse d'un modèle de projection perspective à l'échelle (caméra para-perspective) et permet de se ramener progressivement vers un modèle de projection perspective pure. Cette méthode est extrêmement rapide en temps de calcul grâce à des précalculs judicieux et permet un calcul de la pose beaucoup plus précis que les approches analytiques ou linéaires. Une solution similaire dans le cas des droites est proposée dans [34].

Orthogonalement à cette première classification il est possible de considérer les primitives géométriques utilisées pour l'estimation de la pose. Ce sont, le plus souvent, des points [23, 65, 64, 83, 76, 56, 118] mais on trouve aussi des segments [58, 34, 116, 52], des contours [117, 60, 142], des coniques [172, 53], ou des objets cylindriques [57]. Peu d'approches considèrent l'utilisation conjointe de plusieurs types de primitives (voir cependant [163] pour l'utilisation de points et de droites ou *[134]*). Pour la calibration, là encore, la grande majorité des méthodes existantes repose sur l'extraction de points (e.g., [64, 23, 196, 203, 202]), mais il existe des approches utilisant des droites [25] ou des ellipses [53, 189].

5.3 Suivi hybride 2D-3D

Si la contrainte issue d'un recalage 2D-3D est très forte et permet souvent d'assurer un suivi robuste des objets considérés, ce n'est cependant pas la seule contrainte envisageable. En supposant que les objets suivis sont rigides, il est possible de considérer dans le suivi

les contraintes issues de l'analyse du mouvement 2D apparent des objets entre deux images consécutives. Il est en effet possible dans certaines conditions de relier le mouvement dans l'image au mouvement de la caméra.

Si pour des objets plans un modèle de déplacement homographique permet de prendre en compte exactement les déplacements 3D de la cible, ce n'est pas le cas pour des objets quelconques à cause des effets de perspective qui ne peuvent pas être pris en compte par ce type de modèle. Une seconde phase est donc nécessaire pour prendre en compte ce type de déformations apparentes. Deux approches peuvent être considérées. Lorsque l'on s'intéresse uniquement au suivi de l'objet dans la séquence d'images, une solution consiste à utiliser des modèles approximatifs et à n'exploiter que des modèles d'objets et de mouvements 2D. Il est alors nécessaire d'introduire une composante déformable pour pouvoir s'adapter aux déformations non linéaires des projections de l'objet dans le plan image liées à des effets de perspective non pris en compte par ces modèles 2D [90, 160]. Le défaut de ce type d'approche réside principalement dans le fait que l'hypothèse de rigidité sous-jacente à l'utilisation de mouvement 2D apparent n'est plus assurée dès lors que l'on introduit une composante déformable dans les objets.

Nous avons donc préféré coupler l'estimation du mouvement apparent à une seconde étape de recalage reposant sur l'utilisation d'un modèle 3D de l'objet. Cette méthode enchaîne deux étapes de transformation globale, la première à caractère 2D, la seconde à caractère 3D. Un premier recalage de la silhouette suivie, pouvant appréhender des grands déplacements, est réalisé via l'estimation d'un modèle affine 2D de mouvement à l'aide d'une méthode robuste. Cette dernière prend en entrée les composantes perpendiculaires au contour du déplacement des points de la projection des arêtes du modèle de l'objet, composantes calculées par la méthode des ECM décrite dans le paragraphe 5.1. Nous pouvons ensuite évaluer les paramètres d'une transformation 3D par la minimisation itérative d'une fonction de coût non linéaire par rapport aux paramètres de pose (calculer la pose consiste à calculer la position et l'orientation de la caméra dans le repère de la scène). Elle consiste à positionner au mieux la projection des arêtes du modèle CAO de l'objet sur les contrastes d'intensité dans l'image avec prise en compte de l'orientation de ces derniers. On utilise alors explicitement dans le suivi la propriété de rigidité des objets.

Estimation de la transformation 2D. Nous considérons dans un premier temps que la transformation entre deux projections successives de l'objet dans le plan image peut être représentée par un modèle de déplacement homographique 2D. L'objectif de cette première étape est d'estimer les paramètres de transformation 2D même en cas de grands déplacements de l'objet. Contrairement aux méthodes reposant sur l'utilisation du filtrage de Kalman pour prédire les positions successives de l'objet [70, 94, 100], cette méthode ne requiert pas l'introduction d'un modèle d'état (par exemple, un modèle à vitesse ou accélération constante), ni l'initialisation, souvent problématique, des matrices de covariance des bruits associés aux modèles d'état et de mesure.

Pour une ensemble de k points $\mathbf{p}_i^t$ et leur correspondant $\mathbf{p}_i^{t+1}$ nous pouvons estimer la transformation homographique 2D $\mathbf{H}$ (telle que pour tout point de l'objet on ait $\mathbf{p}_i^{t+1} = \mathbf{H}\mathbf{p}_i^t$). Il est possible de définir la composante $\mathbf{d}_i^{\perp}$ du mouvement orthogonal au contour

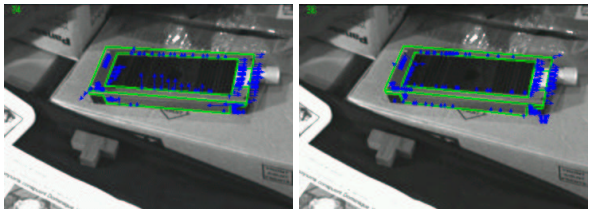


FIG. *5.4: Calcul des déplacements normaux aux contours pour deux images successives*

projeté (voir figure 5.4) :

$$\mathbf{d}_i^{\perp} = \mathbf{n}_i^T \big(\mathbf{H}\mathbf{p}_i - \mathbf{p}_i\big) \tag{5.2}$$

où $\mathbf{n}_i$ est un vecteur unitaire orthogonal à l'arête projetée de l'objet au point $\mathbf{p}_i$ (exprimé en coordonnées homogènes). Partant de (5.2), nous pouvons utiliser un estimateur robuste (un M-estimateur ρ) pour obtenir $\mathbf{H}$ [159]:

$$\widehat{\mathbf{H}} = \arg\min_{\mathbf{H}} \sum_{i=1}^{k} \rho\big(\mathbf{d}_i^{\perp} - \mathbf{n}_i^T \big(\mathbf{H}\mathbf{p}_i - \mathbf{p}_i\big)\big)$$

Cette approche robuste permet de ne pas être affecté par la présence d'éventuelles mesures locales $\mathbf{d}_i^{\perp}$ incorrectes (dues aux ombres, à des erreurs de mise en correspondance, à des occultations, etc.).

Suivi 3D : calcul de pose. Connaissant la position des arêtes projetées de l'objet à l'instant t et l'estimation $\widehat{H}$ de la transformation homographique entre les instants t et $t+1$, il est possible de calculer la position correspondante de l'objet à l'instant $t+1$. Une transformation homographique ne prend cependant pas en compte complètement les transformations subies par la projection de l'objet (effet de perspective, rotations importantes, etc.). Après quelques itérations, le procédé de suivi peut alors être mis en échec.

Connaissant la position 2D de l'objet déplacé et son modèle 3D, il est possible de calculer une pose en utilisant, par exemple [56]. Nous obtenons alors une première évaluation $\mathbf{r}_{init}^{t+1}$ des paramètres de pose qui doit être affinée pour correspondre le plus précisément possible au nouvel aspect de l'objet. Cette étape consiste à recaler la projection du modèle sur les gradients d'intensité de l'image. Ceci est réalisé par une minimisation itérative d'une fonction d'énergie non linéaire, fonction de $\mathbf{r}$, avec $\mathbf{r}_{init}^{t+1}$ comme valeur d'initialisation.

Plus précisément, nous estimons les paramètres de pose $\widehat{\mathbf{r}}$ tels que $\widehat{\mathbf{r}} = \arg\min_{\mathbf{r}} E(\mathbf{r})$ où la fonction d'énergie $E(\mathbf{r})$ est définie par :

$$E(\mathbf{r}) = \int_{\Gamma_{\mathbf{r}}} \frac{|\,\nabla I(\mathbf{p}).\mathbf{n}\,|}{\|\nabla I(\mathbf{p})\|^2} d\mathbf{p} \tag{5.3}$$

où $\Gamma_{\mathbf{r}}$ représente la partie visible des arêtes projetées du modèle de l'objet pour la pose $\mathbf{r}$, $\mathbf{p}$ est la projection d'un point de $\Gamma_{\mathbf{r}}$ dans l'image, $\nabla I(\mathbf{p})$ représente le gradient spatial de l'intensité lumineuse au point $\mathbf{p}$ le long de l'arête projetée et $\mathbf{n}$ est un vecteur unitaire

normal à la courbe en un site $\mathbf{p}$. Pour la bonne pose $\mathbf{r}$, le produit scalaire $\nabla I(\mathbf{p}).\mathbf{n}$ doit être nul. Précisons que pour obtenir de bons résultats et un temps de calcul efficace, il faut faire très attention à la manière de discrétiser $\Gamma_{\mathbf{r}}$ en un ensemble de point $\mathbf{p}$ et à la minimisation de la fonction d'énergie E *[128]*.

Cette étude, menée à la demande de la Direction des études et recherche d'Électricité de France[1], a permis le développement d'un algorithme relativement rapide permettant le suivi d'objet polyédrique dans des séquences d'images. Quelques résultats sont présentés sur la figure 5.5.

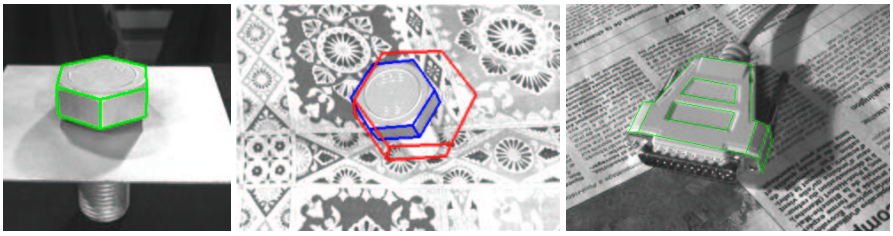


FIG. *5.5: Quelques exemples d'objets suivis par l'approche mixte 2D-3D* [128]*. Le suivi se faisait (en 1998) à une cadence de 5 à 10Hz suivant les objets.*

5.4 Asservissement visuel virtuel

Si l'utilisation d'un modèle 3D de l'objet introduit indéniablement une très forte contrainte sur le processus de recalage, le processus décrit dans le paragraphe précédent reste très perfectible. La formulation de la fonction d'énergie (5.3) ne permet pas d'utiliser des algorithmes classiques mais efficaces de minimisation d'équations non linéaires (de type Gauss-Newton ou Levenberg-Marquardt). Il est en effet quasiment impossible d'obtenir une formulation analytique du Jacobien qui relie la variation de E à la variation de la pose. Il reste que le fait de résoudre le problème du suivi en résolvant le problème de la localisation 3D de l'objet par rapport à la caméra en utilisant une seule image (le calcul de pose) nous semble être une approche efficace.

Nous nous sommes donc intéressés à ce problème de calcul de pose en revenant à sa formulation initiale à savoir comme un problème de recalage consistant à déterminer la relation existant entre un ensemble de primitives 3D et leurs projections 2D dans le plan image. Dans nos travaux, nous avons considéré ce calcul de pose comme un problème de recalage non-linéaire global : l'asservissement visuel virtuel (AVV) [183]*[134]*. Il est en effet possible de considérer que l'asservissement visuel est le problème dual du calcul de pose. Résoudre l'équation (5.1) consiste à trouver un modèle de caméra virtuelle ayant les mêmes carac-

[1]Afin de réduire les risques encourus par les intervenants humains en environnements sensibles (centrales nucléaires, travaux sous tensions), les opérations de maintenance courantes de EdF serait amenées à être de plus en plus souvent effectuées par des systèmes semi ou complètement automatisés. L'asservissement visuel constituait l'une des voies explorées par EdF pour atteindre ce but.

téristiques que la caméra réelle. Pour résoudre ce problème il suffit donc de déplacer une caméra virtuelle afin de minimiser l'erreur entre la projection des informations visuelles sur son plan image et les observations dans l'image réelle en utilisant une loi de commande d'asservissement visuel. Considérer ainsi le calcul de pose permet d'utiliser toutes les techniques issues de l'asservissement visuel (par exemple, les matrices d'interaction associées à un grand nombre de primitives sont parfaitement connus). De plus il est aisé de considérer dans le même processus d'estimation différentes primitives géométriques.

Notons que cette formulation initialement proposé dans [183] a ensuite été reprise dans nos travaux *[133, 134]*. Une formulation très similaire a aussi été proposée à la même période dans [142].

5.4.1 Asservissement visuel virtuel : Principe

L'idée fondamentale de notre approche est donc de définir le calcul de pose comme le problème dual de l'asservissement visuel 2D. Comme nous l'avons expliqué dans les premières parties de ce document, l'objectif de l'asservissement visuel est de déplacer une caméra afin d'observer un objet à une position donnée dans l'image. Ceci est réalisé en minimisant l'erreur entre la position désirée des informations visuelles $\mathbf{s}^*$ dans l'image et leur position courante $\mathbf{s}$. Si le vecteur des informations visuelles est bien choisi, il y a une unique position de la caméra qui assure la régulation à zéro de cette erreur. Nous décrivons maintenant pourquoi le calcul de pose est un problème très similaire.

Pour illustrer notre propos, et sans perte de généralité, nous considérerons ici, le cas d'un objet constitué de N points. L'objectif du calcul de pose est de minimiser l'erreur donnée par l'équation (5.1) entre les données observées $\mathbf{p}_i$ et notées $\mathbf{s}^*$ (comme en asservissement visuel, on a $\mathbf{s}^* = (\mathbf{p}_1, \ldots, \mathbf{p}_N)^T$) et la position des mêmes informations visuelles $\mathbf{s}$ calculée par projection en fonction des paramètres extrinsèques ${}^c\mathbf{M}_o$ (le vecteur $\mathbf{s}$ est donnée par $\mathbf{s} = (\mathbf{K^c M_o{}^o P}_1, \ldots, \mathbf{K^c M_o{}^o P}_N)^T$). Par rapport à un asservissement classique où $\mathbf{s}$ est extraite de l'image et $\mathbf{s}^*$ est définie soit par calcul soit par apprentissage, dans le cas de l'asservissement visuel virtuel, $\mathbf{s}$ est obtenue par un calcul de reprojection sur le plan image d'une caméra virtuelle dont la position courante est ${}^c\mathbf{M}_o$ et $\mathbf{s}^*$ est mesurée dans l'image. De manière à réaliser la minimisation de l'erreur $\|\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\|$, nous déplaçons la caméra virtuelle (initialement en ${}^{ci}\mathbf{M}_o$) en utilisant une loi de commande classique d'asservissement visuel et donnée par l'équation (3.5) (voir les détails dans la section 2.1). Quand la minimisation est réalisée, la position finale de la caméra est donnée par ${}^{cf}\mathbf{M}_o$ c'est-à-dire la pose. Cet exemple est illustré ici pour des points. Pour d'autres types de primitives géométriques les équations de projection et de changement de repère sont bien évidemment différentes mais le principe général reste identique. Ce processus est illustré par la figure 5.6.a en considérant des droites comme information visuelle. Sur la figure 5.6.b des points sont considérés et la méthode est étendue au processus de calibration.

Comme nous le verrons par la suite, ces techniques d'asservissement visuel virtuel peuvent être adaptées à de nombreux problèmes de vision par ordinateur se formulant sous la forme de minimisation d'un système d'équations non-linéaires : calibration, estimation d'homographies, etc. En fonction du modèle choisi pour la matrice d'interaction $\widehat{\mathbf{L_s}}$, on peut en effet montrer l'équivalence de cette méthode de minimisation avec des approches de type Gauss-Newton, Levenberg-Marquardt (équivalente aux lois de commande de type

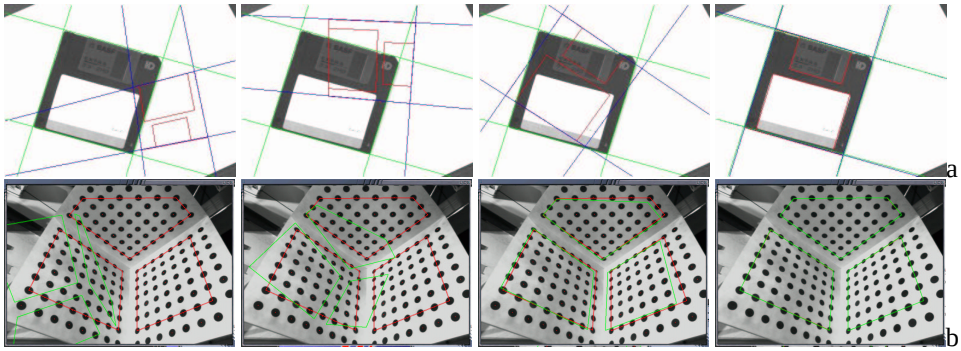


FIG. *5.6: Pose et calibration par asservissement visuel virtuel: principe. L'objectif est de modifier itérativement la position de la caméra en utilisant une loi de commande classique d'asservissement visuel afin de recaler les primitives visuelles extraites de l'image avec les primitives calculées par projection du modèle 3D. Sur la première ligne, des droites sont utilisées pour calculer la pose. Sur la seconde ligne des points sont utilisés pour calculer à la fois la pose et les paramètres intrinsèques de la caméra.*

Damped Least Square Method [200, 121] ou même Newton [121] (une analogie entre les lois de commande référencées capteurs en robotique et les problèmes de minimisation non linéaire a été très récemment présentée dans [121]).

N'importe quel type de primitives géométriques peut être considéré dans la loi de commande proposée dès lors que nous pouvons calculer la matrice d'interaction associée à $\mathbf{L_s}$. Dans [62], un cadre général est proposé pour calculer $\mathbf{L_s}$. C'est un des avantages de cette approche par rapport à d'autres approches non-linéaires de calcul de pose. En effet nous pouvons calculer la pose à partir d'un grand nombre d'informations visuelles différentes (points, lignes, cercles, quadriques, distances, etc.). Il est également très facile de mélanger différentes primitives en ajoutant des primitives au vecteur $\mathbf{s}$ et en empilant les matrices d'interaction correspondantes. En outre, si le nombre ou la nature des primitives visuelles sont modifiés avec le temps (dans un contexte de suivi), la matrice d'interaction $\mathbf{L_s}$ et le vecteur d'erreur $\mathbf{s}$ peuvent être modifiés en conséquence. Dans *[134]*, nous avons appliqué cette approche en considérant les primitives visuelles classiquement utilisées en asservissement visuel (voir résultat sur la figure 5.7).

5.4.2 Asservissement visuel virtuel robuste

Le fait que l'information visuelle issue de l'image doive être calculée avec une précision suffisante est une hypothèse importante. De nombreux travaux ont considéré la présence de données aberrantes dans le processus de calcul de pose. Dans ce contexte, il s'agit principalement d'erreur de mise en correspondance entre les informations 2D et leurs homologues 3D. Comme nous l'avons déjà évoqué dans la section 3.3 deux groupes principaux d'algorithmes robustes permettent de prendre en compte ces données aberrantes (*outliers*). La première approche consiste à détecter les *outliers* avant de procéder à l'estimation des paramètres. L'approche de ce type la plus classiquement utilisée est l'algorithme Ransac [65]. Cet algorithme consiste à estimer les paramètres recherchés avec le minimum de mesures nécessaires puis à vérifier si d'autres mesures confirment cette première estimation. Si un

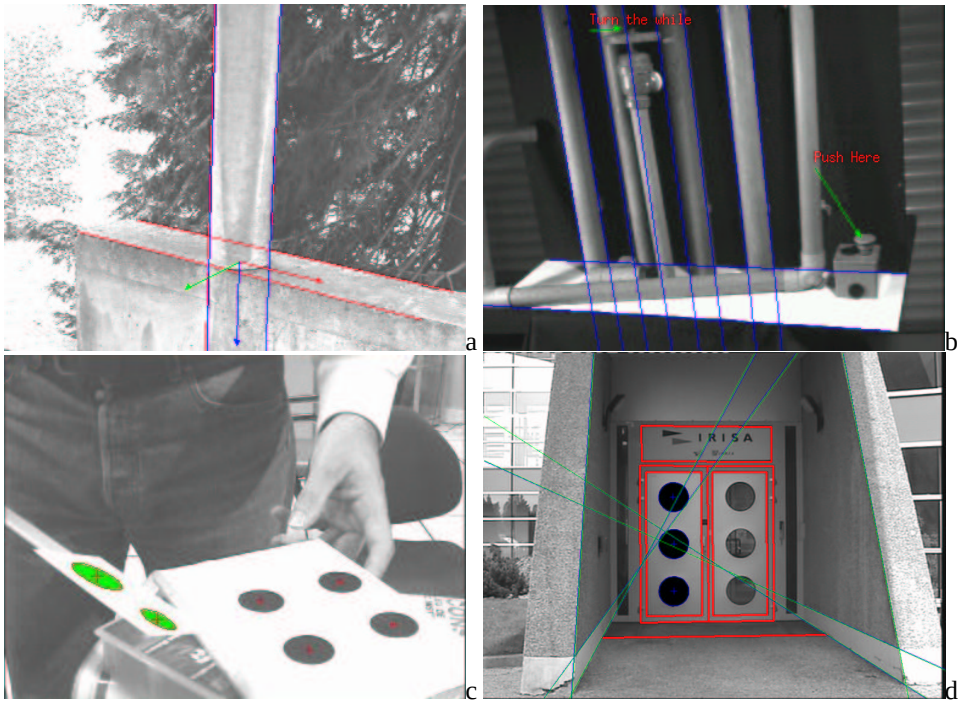


FIG. *5.7: Calcul de pose réalisé à partir de plusieurs primitives de nature différente (a-b) droite et cylindre (c) cercle et points (d) cercles et droites*

consensus est obtenu, l'estimation est retenue. La seconde approche permet de résoudre simultanément le problème de la détection des *outliers* et de l'estimation (e.g., LMedS [169], M-estimateurs, L-Estimateurs ou R-Estimateurs [86]). Ces techniques visent à redéfinir la fonction d'objectif à minimiser afin que le minimum global de la fonctionnelle ne soit pas affecté par les données aberrantes. Ces approches cherchent par ailleurs à estimer de façon robuste l'écart type des "bonnes" mesures ou des mesures non aberrantes (à la différence de Ransac qui considère cet écart type comme une constante connue). Le lecteur pourra se référer à [181] pour une analyse des différentes techniques d'estimation robuste appliquées à la vision par ordinateur et à [106] pour l'utilisation des M-estimateurs et de l'approche LMedS pour le calcul de pose.

Dans le cadre de la thèse d'Andrew Comport, nous avons retenu l'utilisation de M-estimateurs pour parvenir à cet objectif. En asservissement visuel, la loi de commande qui réalise la minimisation de Δ est traitée habituellement par une approche au moindres carrés (voir section 2.1). Cependant, si il y a des données aberrantes, une estimation robuste est nécessaire. Comme nous l'avons vu dans la section 3.3 ce problème peut être résolu en considérant une loi de commande robuste[2]. Cette loi de commande reposant sur l'utilisa-

[2]Notons que les lois de commande d'asservissement visuel robuste décrites dans la section 3.3 ont été initialement développées dans le cadre de cet asservissement visuel virtuel puis transposées dans un contexte de

tion de M-estimateurs a pour objectif de minimiser non plus l'erreur Δ mais une fonction robuste de cette erreur afin de réduire la sensibilité aux données aberrantes. L'équation d'optimisation robuste est donnée par :

$$\Delta_{\mathcal{R}} = \sum_{i=1}^{N} \rho\big(s_i(\mathbf{r}) - s_i^*\big)^2, \tag{5.4}$$

où $\rho(u)$ est une fonction robuste [86] et la loi de commande résultante est donnée par :

$$\mathbf{v} = -\lambda(\widehat{\mathbf{D}}\widehat{\mathbf{L}_\mathbf{s}})^+\mathbf{D}\big(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*\big). \tag{5.5}$$

Le calcul des poids représentant la confiance en chaque primitive a été présenté dans la section 3.3 et la même approche s'applique ici.

Comme dans le cas de l'asservissement visuel, le choix du modèle pour $\widehat{\mathbf{D}}$ et pour $\widehat{\mathbf{L}_\mathbf{s}}$ est important et comme pour une loi de commande "standard" une analyse de stabilité et de convergence peut être réalisée. Le choix classique en asservissement visuel $\big[\widehat{\mathbf{D}}\widehat{\mathbf{L}_\mathbf{s}}\big]^+ = \big[\mathbf{L}_\mathbf{s}(\mathbf{s}^*, \mathbf{r}^*\big]^+$ est ici impossible puisque si $\mathbf{s}^*$ est connu, $\mathbf{r}^*$ (la pose recherchée), est inconnue. Dans un processus de suivi, l'erreur dans l'image étant relativement faible, le choix $\big[\widehat{\mathbf{D}}\widehat{\mathbf{L}_\mathbf{s}}\big]^+ = \big[\mathbf{D}(\mathbf{s_i})\mathbf{L}_\mathbf{s}(\mathbf{s_i}, \mathbf{r}_i)\big]^+$ où $\mathbf{r}_i$ est la pose initiale de la caméra virtuelle et $\mathbf{s_i}$ la valeur initiale des primitives visuelles est intéressant puisque $(\widehat{\mathbf{D}}\widehat{\mathbf{L}_\mathbf{s}})^+$ est calculée seulement une fois. D'autre choix sont cependant possibles *[36, 38]*.

5.4.3 Suivi d'objets complexes en temps-réel

Dans les sections 5.4.1 et 5.4.2 nous avons considéré le problème du calcul de pose mono-image. Cette méthode s'adapte parfaitement au suivi d'objets temps-réel dans une séquence vidéo. On se retrouve dans la classe des approches de suivi basé modèle [117, 60, 142, 198]. L'avantage principal des méthodes basées sur un modèle 3D de l'objet est que la connaissance *a priori* sur la scène (l'information 3D implicite) permet l'amélioration de la robustesse et de la performance tout en étant capable de fournir l'information supplémentaire nécessaire pour réduire les effets de données aberrantes présentes éventuellement dans le processus de suivi. De plus, s'il est possible de résoudre efficacement le problème de la localisation 3D à une fréquence élevée (25 ou 50 Hz), l'amplitude du déplacement des objets dans l'image est alors très faible et l'appariement des primitives visuelles 2D (et donc le suivi) est plus fiable.

L'une des difficultés induites par de tels algorithmes porte sur l'efficacité et la précision avec lesquelles les primitives locales de l'image sont combinées avec le modèle global de l'objet. Ceci pose donc d'une part le problème de la nature de l'information visuelle $\mathbf{s}$ utilisée dans la loi de commande et donc de la modélisation de la matrice d'interaction $\mathbf{L}_\mathbf{s}$ associée et, d'autre part, le problème de l'extraction de cette information visuelle $\mathbf{s}$. Dans le cadre de la thèse d'Andrew Comport, notre choix s'est porté sur des primitives de type distance orthogonale d'un point 2D $\mathbf{p}$ à la projection du contour dans l'image. Dans ce cas particulier où la primitive est une distance, la valeur désirée $\mathbf{s}^*$ est égale à zéro. Nous avons fait l'hypothèse que les contours de l'objet peuvent être divisés en segments ou en portions

commande de robots.

d'ellipses. Tous les points qui correspondent à un segment particulier ou une ellipse sont alors traités indépendemment. Dans le cas des segment, la distance entre un point $\mathbf{p}$ observé dans l'image et la projection $\mathbf{l}(r)$ du segment pour une pose $\mathbf{r}$ peut être caractérisée par la distance perpendiculaire $d_\perp$ et cette droite (voir figure 5.8). La distance parallèle au segment ne contient, elle, aucune information utile à moins qu'une correspondance existe entre un point sur la ligne et $\mathbf{p}$ (ce qui n'est pas le cas ici). Nous avons donc :

$$d_\perp = d_\perp(\mathbf{p}, \mathbf{l}(r)) = \rho(\mathbf{l}(r)) - \rho_d, \tag{5.6}$$

où la position de la droite $\mathbf{l}(r)$ est donnée par sa représentation en coordonnées polaires. Grâce au cadre général proposé dans [62] pour calculer $\mathbf{L_s}$, il est possible de calculer la matrice d'interaction liée à $\mathbf{d}_\perp$ *[36]*. Notons que les cas des distances entre un point et la projection d'un cylindre et d'une ellipse sont très semblables. La plupart des approches de ce type (e.g. [116]) considèrent non pas la distance d'un point au contour mais la distance entre un point $\mathbf{p}$ et la projection d'un point du contour $\mathbf{p}(\mathbf{r})$. La matrice d'interaction correspondante est donc celle associée à un point. Il n'y a cependant aucune certitude pour que ces deux points 2D correspondent au même point physique 3D et considérer une telle erreur n'est qu'une approximation localement correcte.

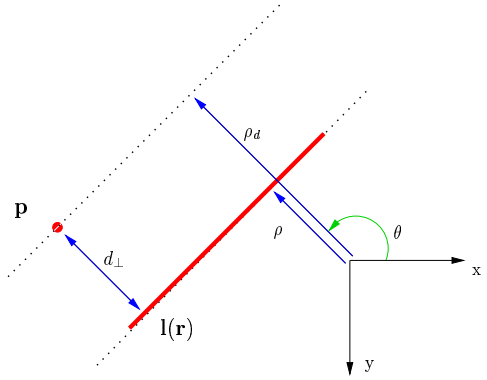


FIG. *5.8: Une des primitives visuelles utilisées dans le processus de suivi : distance d'un point à une droite*

Un second point important concerne l'extraction des informations visuelles de la séquence d'images (c'est-à-dire le calcul de $\mathbf{s}$). Pour cela, les déplacements orthogonaux à la projection du modèle sont évalués en utilisant l'algorithme des éléments de contours en mouvement précédemment décrit dans la section 5.1. Cette approche introduisant une contrainte spatio-temporelle dans l'appariement local est beaucoup plus robuste qu'une simple recherche du gradient maximum dans la direction normale au contour [116, 60, 142, 103] pour un coût calculatoire à peine plus élevé. Un autre avantage de ce processus est qu'il n'exige jamais l'extraction explicite sur l'image de contours [116, 52, 194].

Notons cependant que, dans ce type d'approche, la dimension temporelle est très peu utilisée. On la retrouve, dans le cadre de l'algorithme que nous avons considéré au niveau de l'appariement des indices visuels par les ECM, mais elle est totalement absente du processus de recalage 2D-3D. Il est donc impropre de parler de suivi spatio-temporel si l'on ne

considère pas une contrainte temporelle supplémentaire. Cette contrainte peut reposer sur la géométrie sous-jacente à de tels systèmes (e.g., contrainte épipolaire) ou sur l'utilisation de filtres de Kalman étendus (EKF) [99, 194].

Cet algorithme a été largement utilisé à l'IRISA dans le cadre d'applications d'asservissement visuel. En effet, outre sa robustesse aux occultations (grâce à l'utilisation des M-estimateurs dans la loi de commande), aux variations d'éclairage (robustesse des ECM), à des mouvements "aléatoires" (non compatibles avec l'utilisation efficace d'une étape de prédiction de type Kalman), il permet la localisation 3D et donc 2D à une cadence compatible avec une loi de commande en boucle fermée. Dans les figures 5.9 et 5.10 des objets polyédriques ont été placés dans un environnement fortement texturé. Des tâches de positionnement par asservissement visuel 2D 1/2 [122] (figures 5.9 et 5.10) ont été réalisées. Plusieurs occultations partielles (main, outils, etc) ainsi que des variations d'éclairage importants ont été provoquées pendant la réalisation de ces tâches de positionnement. Dans tous les cas, le suivi est toujours exécuté à une cadence compatible avec la cadence vidéo (souvent en moins de 10 ms). D'autres types de distances ont été utilisés dans le contexte applicatif de la réalité augmentée, les résultats seront présentés dans la section 5.6.

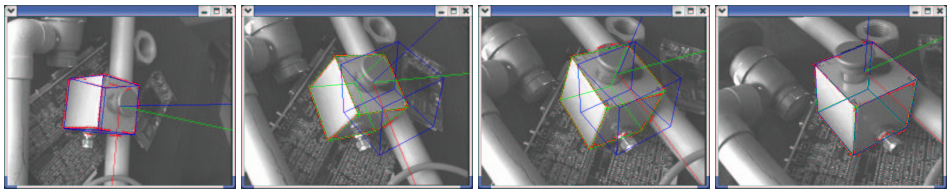


FIG. *5.9: Suivi d'un objet pendant une expérience d'asservissement visuel 2D 1/2. L'objet suivi est en vert et sa position désirée dans l'image est en bleu. Les points rouges correspondent aux points suivis par l'algorithme des ECM*

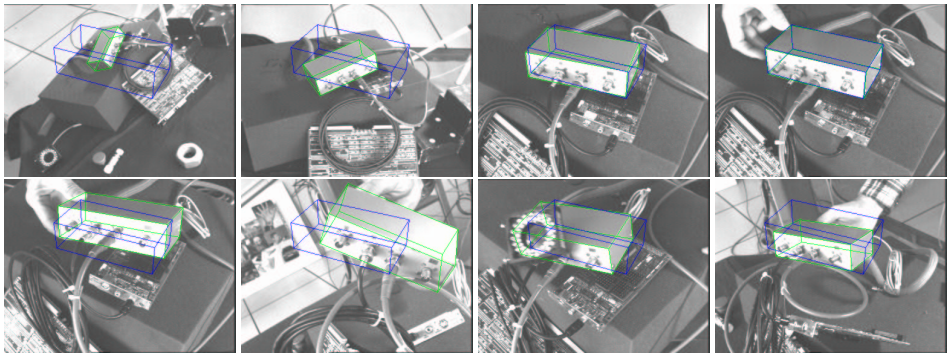


FIG. *5.10: Suivi d'un objet pendant une expérience d'asservissement visuel 2D 1/2. L'objet suivi est en vert et sa position désirée dans l'image est en bleu. Les images de la première ligne correspondent à l'étape initiale de positionnement. Dans la suivante, à la fois l'objet et le robot sont en mouvement.*

5.5 Autres utilisations possibles pour l'asservissement visuel virtuel

Si sa formulation en terme de loi de commande est différente, l'asservissement visuel virtuel est une approche très similaire à celle reposant sur des algorithmes itératifs de minimisation de systèmes d'équations non linéaires. La solution d'un grand nombre de problèmes classiques en vision par ordinateur peut être obtenue en résolvant de tels systèmes : pose, étalonnage, estimation de la géométrie épipolaire, *structure from motion*, reconstruction 3D d'objets paramétriques, etc.

Même si une solution est envisageable, l'utilisation de l'asservissement visuel virtuel n'est sans doute pas toujours optimale. Si la fonction d'erreur à minimiser n'est pas directement exprimée en fonction d'informations 2D issues des images, l'intérêt n'est sans doute plus évident. Afin de monter la généralisation possible de cette formulation, nous avons considéré deux autres problèmes classiques en vision par ordinateur : l'étalonnage des caméras et l'estimation du mouvement d'une caméra (via l'estimation d'une homographie).

5.5.1 Étalonnage de caméra

Il est possible de réaliser l'étalonnage des caméras en utilisant la même approche. Pour la pose, nous avions considéré que le vecteur d'informations visuelles $\mathbf{s}$ était exprimé dans l'espace métrique. Pour la calibration, les paramètres intrinsèques de la caméra étant inconnus, $\mathbf{s}$ et $\mathbf{s}^*$ seront exprimés en pixel et la fonction d'objectif à minimiser est donnée par :

$$\left({}^{\mathbf{c}}\mathbf{M_o}, \widehat{\xi}\right) = \arg\min_{{}^{\mathbf{c}}\mathbf{M_o},\xi} \sum_{i=1}^{N} \left(\mathbf{p}_i - \mathbf{K}(\xi){}^{c}\mathbf{M}_o{}^{\mathbf{o}}\mathbf{P}_i\right)^2 \tag{5.7}$$

où ξ représente les paramètres intrinsèques de la caméra (point principal, rapport mètre pixel, etc) et où la matrice $\mathbf{K}(\xi)$ est la matrice de projection perspective intégrant la transformation mètre/pixel. Dans ce cas l'erreur à minimiser est donnée par :

$$\mathbf{e} = \mathbf{s}(\mathbf{r}, \xi) - \mathbf{s}^* \tag{5.8}$$

Le déplacement des informations visuelles dans l'image est donc relié non seulement au déplacement de la caméra $\mathbf{v}$ mais aussi à la variation temporelle des paramètres intrinsèques $\dot{\xi}$:

$$\dot{\mathbf{s}} = \underbrace{\begin{pmatrix} \mathbf{L_s} & \dfrac{\partial \mathbf{s}}{\partial \xi} \end{pmatrix}}_{\mathbf{H}} \begin{pmatrix} \mathbf{v} \\ \dot{\xi} \end{pmatrix} \tag{5.9}$$

La figure 5.11 montre le résultat de cette minimisation dans le cas d'un objectif grand-angle de 6mm (une première étape de calibration par un algorithme linéaire (Toscani-Faugeras [64]) avait fourni une initialisation à notre algorithme). Des comparaisons avec d'autres approches de calibration sont disponibles dans *[133]*.

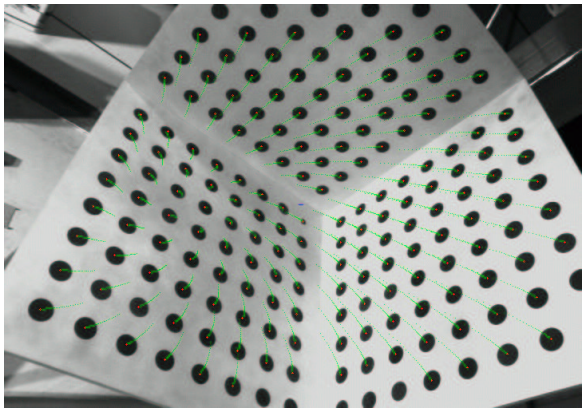


FIG. *5.11: Calibration d'un objectif de 6mm par asservissement visuel virtuel. La position des points en vert correspond à la reprojection des points du modèle 3D pour les paramètres de calibration estimés.*

Les paramètres intrinsèques issus d'une calibration mono-image peuvent être, en pratique, très différents des paramètres calculés à partir d'une seconde image prise d'un autre point de vue et ce avec la même caméra, le même objectif, la même carte de numérisation, et la même mire de calibration. Ceci est en grande partie dû à la très forte corrélation entre les paramètres intrinsèques et extrinsèques et principalement entre le zoom et la translation en z. Cette approche est donc difficilement utilisable en ligne. Il est cependant possible de considérer plusieurs images dans le processus de calibration et d'intégrer l'ensemble des données disponibles dans le processus d'estimation [111] et l'asservissement visuel virtuel se prête efficacement à ce genre d'extensions *[133]*.

5.5.2 Estimation du déplacement d'une caméra

Le calcul du déplacement de la caméra entre deux positions peut se faire de façon assez similaire au calcul de pose. Considérons le cas d'une scène composée d'un certain nombre de primitives visuelles 2D $\mathbf{s}$ (par exemple des points, des distances, etc). Pour estimer le déplacement de la caméra, une approche classique [77] est de minimiser la distance dans l'image entre la position de ces primitives mesurée dans l'image 2 ($\mathbf{s}_2$) et leur position dans l'image 1 ($\mathbf{s}_1$) transférée dans l'image 2 par une transformation particulière ${}^2tr_1(\mathbf{s}_1)$.

$$\widehat{{}^{c_2}\mathbf{M}_{c_1}} = \underset{{}^2\mathbf{M}_1}{\operatorname{argmin}}\ \Delta \ \text{ avec } \ \Delta = \sum_{i=1}^{N} \left(\mathbf{s}_{2_i} - {}^2tr_1(\mathbf{s}_{1_i})\right)^2 + \left(\mathbf{s}_{1_i} - {}^1tr_2(\mathbf{s}_{2_i})\right)^2$$

où N est le nombre de primitives visuelles considérées et ${}^2\mathbf{e}_{1_i}^2 = \mathbf{s}_{2_i} - {}^2tr_1(\mathbf{s}_{1_i})$ est la distance signée entre les primitives 2D $\mathbf{s}_{2_i}$ et ${}^2tr_1(\mathbf{s}_{1_i})$. Afin de prendre en compte des erreurs dans l'extraction des primitives, il est souhaitable de minimiser les erreurs croisées dans les deux images. On considère donc les transformations directes (${}^2tr_1$) et indirecte (${}^1tr_2$).

En formulant ce problème dans le cadre de l'asservissement visuel virtuel *[165]*, une caméra virtuelle effectue un déplacement ${}^{\mathbf{2}}\mathbf{M_1}$ afin de minimiser cette erreur Δ. Minimiser

cette distance est équivalent à minimiser le vecteur d'erreur :

$$\mathbf{e} = \left[{}^2\mathbf{e}_{11}, {}^1\mathbf{e}_{21}, \ldots, {}^2\mathbf{e}_{1i}, {}^1\mathbf{e}_{2i}, \ldots, {}^2\mathbf{e}_{1N}, {}^1\mathbf{e}_{2N}\right] \tag{5.10}$$

par la loi de commande suivante :

$$\mathbf{^2v} = -\lambda \widehat{\mathbf{L}}_{\mathbf{e}}^{+} \mathbf{e} \tag{5.11}$$

où $\mathbf{^2v}$ est la vitesse de la caméra (exprimée dans le repère de la caméra 2) et où $\mathbf{L_s}$ est la matrice d'interaction associée au vecteur d'erreur et définie par :

$$\widehat{\mathbf{L}}_{\mathbf{e}} = \left(\cdots, \quad \widehat{\mathbf{L}}({}^2\mathbf{e}_{1i}), \quad -\widehat{\mathbf{L}}({}^1\mathbf{e}_{2i}) {}^1\widehat{\mathbb{T}}_2, \quad \cdots \right) \tag{5.12}$$

$\mathbf{L}({}^k\mathbf{e}_{li})$ lie la variation de la distance ${}^k\mathbf{e}_{li}$ au mouvement de la caméra virtuelle et dépend évidemment de la transformation retenue. ${}^1\mathbb{T}_2$ est la matrice de changement de repère pour le torseur cinématique.

Dans le cas général (scène non coplanaire et mouvements de caméra quelconques) le transfer de point peut s'effectuer en utilisant plusieurs images et en considérant la géométrie épipolaire et la matrice essentielle ou fondamentale [77]. Nous nous sommes resteint au cas moins général (scène plane ou à l'infinie, mouvement de rotation pure) ou le transfert de point peut être réalisé en utilisant un homographie. Dans ce cas, un point ${}^1\mathbf{p}$ exprimé en coordonnées homogènes ${}^1\mathbf{p} = ({}^1x, {}^1y, {}^1w)$ est transféré en ${}^2\mathbf{p}$ dans l'image 2 en utilisant la relation suivante:

$${}^2\mathbf{p} = {}^2tr_1({}^1\mathbf{p}) = \alpha\, {}^2\mathbf{H}_1 {}^1\mathbf{p} \tag{5.13}$$

où ${}^2\mathbf{H}_1$ est une homographie définie à un facteur d'échelle près. Quand un déplacement de la caméra est généré, l'homographie ${}^2\mathbf{H}_1$ est donnée par [77]:

$${}^2\mathbf{H}_1 = ({}^2\mathbf{R}_1 + \frac{{}^2\mathbf{t}_1}{{}^1d} {}^1\mathbf{n}^T) \tag{5.14}$$

où ${}^1\mathbf{n}$ et ${}^1d$ sont la normale et la distance au plan de référence exprimées dans le repère de la caméra 1. On obtient ainsi ${}^2tr_1 = ({}^2x, {}^2y, {}^2w)$ qui est utilisé pour l'itération suivante de la loi de commande

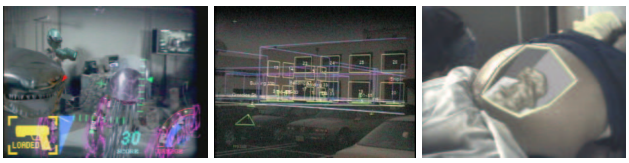
Cette contrainte sur la position spatiale des primitives dans l'image pour un déplacement donné de la caméra constitue en fait une très forte contrainte spatio-temporelle sur la trajectoire suivie par la caméra. Couplée à l'estimation de la pose, elle peut fournir la dimension temporelle absente de la formulation initiale de l'algorithme de suivi proposée dans la section 5.4.3.

5.6 Application à la réalité augmentée temps-réel

Les recherches que nous avons réalisées sur la localisation et le suivi d'objet dans des séquences vidéo ont été initiées par la nécessité de disposer d'outils de traitement d'images

Audiovisuel : cinéma (Terminator 3), publicité (Renault/Realviz),
plateau télévision (France 2, Total-immersion), sport (TF1, Symah-vision)

Industrie : étude de conformité (Siemens)
industrie automobile (Arvika), assemblage (Arvika), étude d'impact (Loria)



Divers : jeux vidéo (Aquagauntlet, MR-lab), militaire (US Marines Corps),
médical (UNC Chapel Hill)

FIG. *5.12: Exemples d'application de la réalité augmentée*

suffisamment robustes et rapides pour être intégrés dans les expérimentations d'asservissement visuel que nous menons à l'IRISA. Il est cependant évident que la robotique n'est pas le seul domaine d'application pour de tels algorithmes. Un domaine applicatif en plein essor reposant en grande partie sur la nécessité de localiser la caméra par rapport à la scène est la *réalité augmentée*.

5.6.1 Problématique

Le concept de réalité augmentée vise à accroître notre perception du monde réel, en y ajoutant des éléments fictifs, non visibles a priori. La réalité augmentée désigne donc les différentes méthodes qui permettent d'incruster de façon réaliste des objets virtuels dans une séquence d'images. Ses applications sont multiples et touchent de plus en plus de domaines (voir figure 5.12) : jeux vidéo et "*edutainment*", cinéma et télévision (post-production, studios virtuels, retransmissions sportives, . . .), industries (conception, design, maintenance, assemblage, pilotage, robotique et télérobotique, implantation, étude d'impact,. . .), médical, etc.

Agrémenter d'objets fictifs une séquence vidéo issue d'un plan fixe ne pose guère de problèmes. Les applications visées demandant souvent énormément de réalisme, il est indispensable que l'ajout d'objets dans une scène ne perturbe pas la cohérence du contenu filmé. Le fait de déplacer la caméra implique cependant un mouvement dans l'image de la

scène filmée. Pour assurer la cohérence entre les deux flux réels et virtuels, un lien rigide doit être maintenu entre les deux mondes. Afin de donner l'illusion que ces objets fictifs appartiennent au même monde, il est nécessaire de bien les placer, bien les orienter et de respecter des facteurs d'échelle par rapport aux objets réellement filmés. Bien placer les objets virtuels par rapport aux objets de la scène nécessite de connaître la position de la caméra par rapport à la scène.

Le problème de la localisation de la caméra est donc important et peut être résolu par diverses approches. On peut utiliser un système de capteurs, comme des capteurs magnétiques qui mesurent la distorsion du champ magnétique pour calculer leur position, des capteurs optiques, des encodeurs sur les moteurs du pied des caméras ou encore, évidemment, le flux vidéo. Cependant, il s'agit ici de se limiter à une approche image, ce qui ramène le problème de réalité augmentée à un problème de vision par ordinateur. Dans certains contextes applicatifs comme le cinéma, l'ensemble de la séquence vidéo est disponible avant le traitement. Dans cette optique de post-production, des traitements lourds en terme de temps de calcul sont envisageables. Des techniques permettant à la fois la reconstruction 3D d'un certain nombre de points de la scène et la localisation 3D de la caméra sont mises en œuvre par des techniques d'autocalibration ou d'ajustement de faisceaux (*bundle adjustment*) . Des logiciels commerciaux reposant sur ce principe sont d'ores et déjà disponibles (on peut citer Boujou de la société 2d3 – issue de l'université d'Oxford – et MatchMover de la société Realviz – issue du projet Robotvis de l'INRIA Sophia Antipolis –). Ces méthodes sont cependant très dépendantes de la qualité de la mise en correspondance des primitives 2D (bruit d'extraction, distribution spatiale, nombre d'erreurs d'appariement,...) et l'utilisateur est parfois mis à contribution.

Dans le cadre d'applications interactives (audiovisuel dans les "conditions du direct", industrie, jeux vidéo interactifs, médical, militaire) le recours à des techniques d'autocalibration n'est pas possible. Des techniques permettant la localisation de la caméra à partir de l'image courante (et éventuellement des précédentes) sont nécessaires [7, 8, 153]. Si un modèle de la scène (ou d'une partie de celle-ci) est disponible, le calcul de points de vue est évidemment une solution idéale à ce problème. Dans le cas où la structure 3D de la scène est (partiellement) inconnue d'autres approches reposant, par exemple, sur le calcul du déplacement de la caméra sont envisageables [179]. Les avantages de ces approches interactives sont multiples :

- Elles permettent une intégration réelle-virtuelle en temps réel (i.e., à la cadence vidéo) car les calculs sous-jacents sont relativement peux coûteux
- Il n'est pas non plus nécessaire de faire un étalonnage "lourd" du système comme c'est le cas si on utilise d'autres types de capteurs (par exemple, des capteurs magnétiques Polhemus ou autre), ni de disposer *a priori* de la séquence complète.
- Elle peut fonctionner sur des plates-formes PC standards ce qui implique un coût relativement faible.

Cependant contrairement aux techniques utilisées pour la post-production, et si un grand nombre de prototypes ont été réalisés, et à l'exception notable de ARToolKit[3], il n'existe pas à notre connaissance de systèmes "sur étagère" distribués commercialement.

[3]ARToolKit a été initialement développé par Hirokazu Kato de l'université d'Osaka, et est actuellement supporté par le HIT Lab. (Human Interface Technology Laboratory) de l'université de Washington, et par le HIT Lab NZ de l'université de Canterbury en Nouvelle Zélande.

5.6.2 Réalité augmentée à partir de marqueurs

Le problème de la localisation 3D basée marqueurs peut sembler trivial. Il n'en demeure pas moins que la grande majorité des systèmes de réalité augmentée temps-réel reposent actuellement sur l'utilisation de marqueurs. Le système de ce type le plus utilisé actuellement est sans contexte le système ARToolKit. Le point de vue est calculé par des techniques simples de calcul de pose linéaire [93]. Le succès de cette librairie est principalement dû à la simplicité de la mise en œuvre et au processus simple mais fiable et rapide de détection des marqueurs dans l'image. Des logiciels similaires ont été développés dans le cadre du projet Arvika[4].

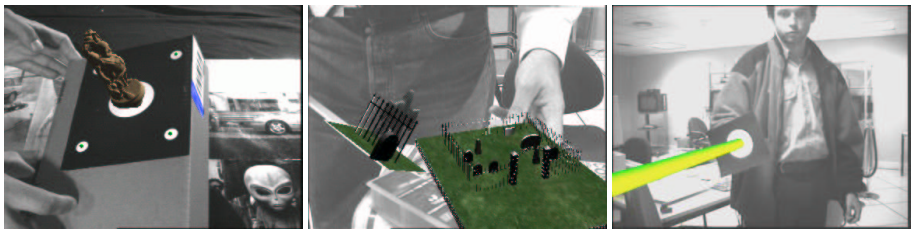


FIG. *5.13: Marker (Marker-based Augmented Reality KERnel) : un logiciel de localisation 3D basée marqueurs pour la réalité augmentée*

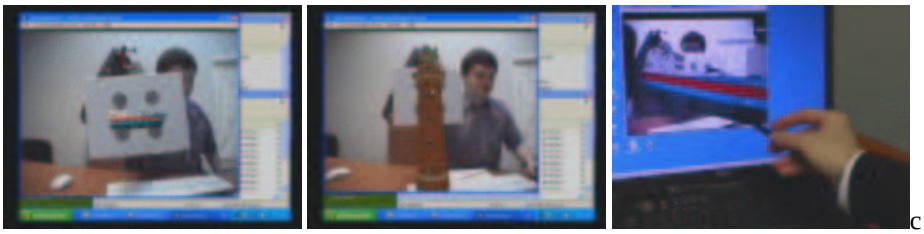


FIG. *5.14: Marker (Marker-based Augmented Reality KERnel) : vidéo-conférence augmentée interactive via un réseau IP (Irisa / Total-immersion)*

Nous avons utilisé les techniques d'asservissement visuel virtuel pour écrire une librairie permettant la localisation et l'incrustation d'objets virtuels en temps réel : Marker (Marker-based Augmented Reality KERnel). Le point de vue peut être calculé à partir de primitives de type points ou cercles (figure 5.13) et l'étalonnage de la caméra peut éventuellement être réalisé en ligne. Ce logiciel est actuellement commercialisé par la société Total-Immersion comme un module de leur système D'Fusion (voir figure 5.14 pour une utilisation en vision conférence augmentée interactive via Internet).

Si de telles approches sont très robustes et permettent le prototypage et la validation rapides de systèmes de réalité augmentée, la présence intrusive de marqueurs dans l'en-

[4]Arvika est un vaste projet sponsorisé par le BMBF (Ministère de la recherche et de l'éducation Allemand) visant à promouvoir la réalité augmentée en environnement industriel et à proposer et mettre en œuvre de tels systèmes (`http://www.arvika.de`). Ce projet est sans doute la raison principale de la très forte implication des milieux industriels et académiques allemands dans le domaine de la réalité augmentée.

vironnement nous semble être un frein majeur au développement de cette technologie en environnement opérationnel. Seul un nombre limité d'applications comme (sans être exhaustif) le design industriel, le jeu vidéo, la vidéo-conférence peuvent, à notre avis, tirer profit de ces méthodes. Par ailleurs il existe des approches qui après une phase d'apprentissage reposant sur l'utilisation de marqueurs permettent de faire un suivi sans marqueur à partir de points d'intérêt [69].

5.6.3 Réalité augmentée sans marqueur

La suppression de la contrainte imposée par les marqueurs est donc un passage nécessaire pour une industrialisation effective de la réalité augmentée interactive. Cependant comme nous l'avons évoqué à plusieurs reprises dans ce chapitre, si un nombre important d'algorithmes de suivi et de localisation 3D existent dans la littérature (voir section 5.4) ceux-ci ne sont pas toujours, pour des raisons diverses (fiabilité, temps de calcul, type de résultats, ...), compatibles avec une application de réalité augmentée. Certains de ces algorithmes ont cependant été développés ou utilisés dans ce contexte [178, 113, 97]*[36]*

Pour notre part, nous avons évidemment retenu sous le nom de Markerless (MarkerLESS-based Augmented Reality KERnel), l'algorithme d'asservissement visuel virtuel robuste décrit dans les sections 5.4.2 et 5.4.3 couplé au module d'insertion d'objets virtuels évoqué dans le paragraphe précédent *[36]*. Les figures 5.15, 5.16, 5.17, 5.18, et 5.20 montrent quelques séquences augmentées réalisées en utilisant Markerless. Comme nous l'avions déjà évoqué dans les paragraphes précédents, l'un des avantages de cet algorithme est sa robustesse face aux occultations partielles, aux variations d'éclairage, aux mouvements relativement importants de la caméra, etc. Ceci est dû d'une part à un algorithme efficace pour gérer les appariements locaux et d'autre part à l'utilisation d'estimateurs robustes dans le processus de minimisation. Comme pour Marker, Markerless est actuellement en cours d'industrialisation par la société Total-Immersion et nous participons activement à son intégration dans le système D'Fusion.

Des modules supplémentaires permettant de gérer la distorsion radiale, de réaliser un filtrage spatio-temporel basé sur un filtre de Kalman étendu, de faire de la fusion de données multi-capteurs (filtrage de Kalman séquentiel ou parallèle), de gérer les changements d'échelles ont (parmi d'autres) été introduits dans le logiciel.

5.6.4 Réalité augmentée sans modèle

L'utilisation de modèle 3D n'est cependant pas toujours satisfaisante dans le sens où les scènes ne peuvent être augmentées que si elle contiennent des objets dont les dimensions sont connues. Ceci est très restrictif et une autre approche peut être utilisée. L'idée, bien répandue, est d'exploiter les données contenues dans la séquence vidéo pour réussir à localiser la caméra par rapport à la scène et ce sans *a priori* sur la scène elle-même. La séquence vidéo est une suite d'images d'une même scène (du moins entre images voisines) et l'exploitation du contenu commun des images se succédant va permettre de résoudre le problème de localisation. Autrement dit, la méthode va se baser sur l'estimation du mouvement de la caméra entre deux images successives et non plus sur une seule image. La pose de la caméra tout au long d'une séquence vidéo peut être obtenue à partir du déplacement

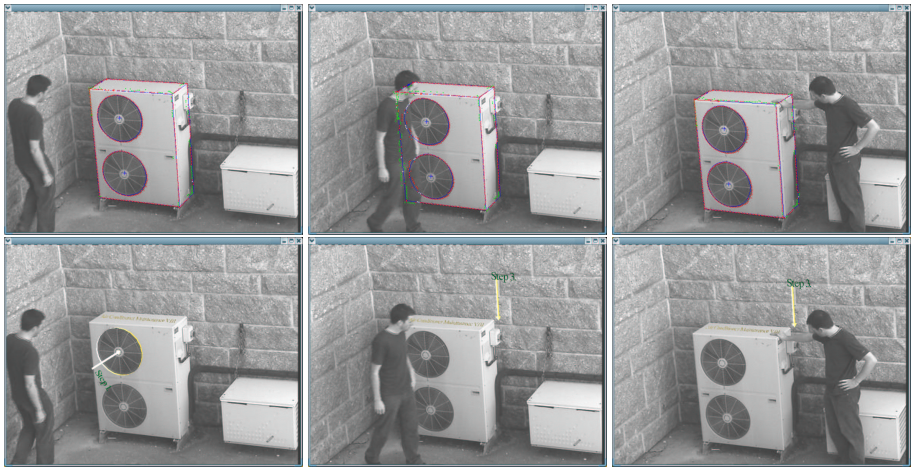


FIG. *5.15: Markerless (Markerless Augmented Reality KERnel) : résistance aux occultations (résultat du traitement d'image et séquence augmentée)*

FIG. *5.16: Markerless : suivi sur fond texturé*

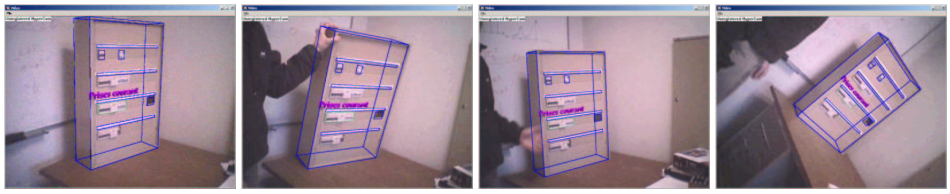


FIG. *5.17: Markerless : suivi d'une armoire électrique, occultations et mouvements importants*

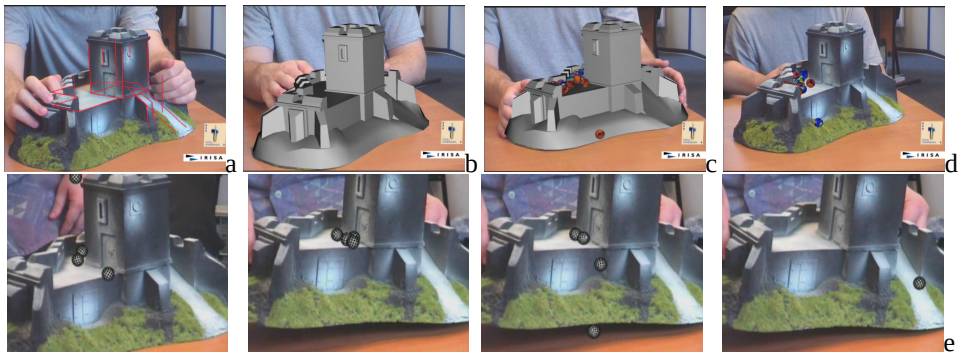


FIG. *5.18: Markerless : suivi d'un château fort, le mouvement des billes incrustées est géré par un moteur physique d'animation intégré dans le logiciel D'Fusion de Total-Immersion. Les images de la première ligne montrent (de droite à gauche) l'objet suivi avec les arêtes saillantes du modèle 3D superposées (a), le modèle 3D complet (b), l'ajout de billes soumises à la gravité et gérées par un moteur physique d'animation (c-d). La seconde ligne (e) correspond à quatre images extraites de la vidéo.*

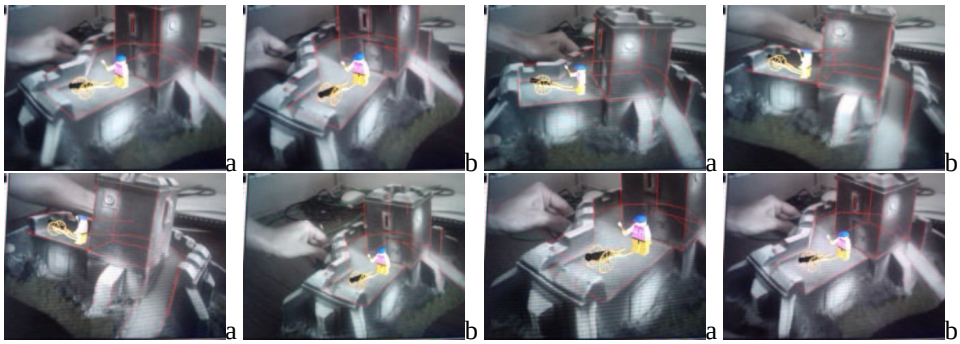


FIG. *5.19: Markerless : autre expérimentation pour le suivi du château fort.*

FIG. *5.20: Markerless : suivi d'une chaise*

de la caméra et de sa pose initiale [180, 179].

L'asservissement visuel virtuel se prête bien à l'estimation des homographies (voir Section 5.5.2) et dans le cas de scènes planes ou de mouvements de rotation pure, il est possible de remonter à une estimation fiable du mouvement de la caméra sur des séquences relativement longues (plus de 1000 images). Sur la séquence présentée sur la figure 5.21, des points caractéristiques (point de Harris) sont extraits et suivis par l'algorithme de Shi-Tomasi [177] et le déplacement de la caméra calculé à une cadence proche de la cadence vidéo (60ms). Les effets de bougé ("*jitter*") sont très faibles et si l'image de référence est régulièrement remise à jour, les dérives sont quasiment inexistantes. Notons que la présence d'un quadrilatère est nécessaire dans la première image pour calculer la pose initiale. Toutefois il n'est pas nécessaire de le maintenir dans le champ de vision tout au long de la séquence.

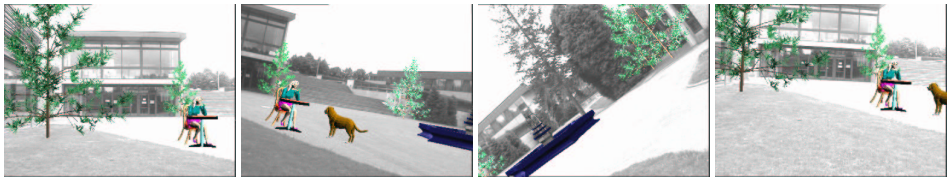


FIG. *5.21: Réalité augmentée sans modèle : estimation en ligne du déplacement de la caméra*

Bilan

L'extraction et l'appariement spatio-temporel de primitives visuelles ou plus généralement d'objets dans des séquences d'images est l'une des difficultés principales qu'il convient de résoudre pour déployer des applications utilisant l'asservissement visuel dans le monde industriel. En 15 ans nous sommes passés d'un suivi de "petits points blanc sur fond noir" réalisé à une cadence de 20ms sur une carte de traitement d'image spécialisée à un suivi d'objets beaucoup plus complexes réalisé en moins de 5ms sur un processeur classique.

Pour un bon nombre d'applications, il est possible de disposer du modèle 3D des objets à considérer. L'introduction de cette connaissance permet de fiabiliser et de robustifier notablement les algorithmes de suivi 2D en les transposant à un problème de localisation et de suivi 3D. Ce problème est classique dans le domaine de la vision par ordinateur. Nous l'avons dans un premier temps abordé en tentant de coupler une estimation du mouvement 2D apparent avec une localisation 3D avant de proposer une méthode reposant sur une formulation du problème sous forme d'asservissement visuel virtuel beaucoup plus efficace. Dans l'avenir nous comptons cependant revenir sur l'utilisation du mouvement afin de considérer une vrai contrainte spatio-temporelle dans le processus de localisation (qui deviendrait alors réellement un suivi...).

CHAPITRE 6

Bilan et perspectives

Ce document met en évidence nos principales contributions au domaine du contrôle du mouvement des caméras (au sens le plus large possible). Il peut paraître surprenant à la lecture des chapitres précédents de voir cohabiter des résultats portant sur la robotique, l'animation par ordinateur, la vision par ordinateur et la réalité augmentée. Cependant, si le contexte de ces différentes études et les problématiques sous-jacentes peuvent sembler éloignés, nous avons tenté d'utiliser dans tous les cas un formalisme commun reposant sur l'asservissement visuel.

Pour chacune de nos contributions décrites dans ce document, nous avons déjà effectué un bilan du travail réalisé. Nous ne reprenons donc ici que brièvement certains des points déjà énoncés avant de proposer quelques perspectives qu'il sera souhaitable d'explorer à court ou moyen terme.

Robotiques et asservissement visuel. Nos activités de recherche en asservissement visuel dans un contexte robotique ont principalement porté sur la définition de stratégies permettant de pallier le défaut intrinsèque de ce type d'approche dans des contextes applicatifs réalistes. Ces défauts proviennent entre autres de l'aspect bas niveau de la commande et de sa sensibilité au bruit dans les mesures. Notre modeste contribution au domaine de l'asservissement visuel se situe donc tant au niveau de la modélisation des tâches à réaliser que de l'adaptation de la commande "classique" aux difficultés que nous venons d'évoquer. Nos perspectives sur ce sujet sont variées et portent à la fois sur des extensions des travaux déjà réalisés mais aussi, à plus long terme, sur des thématiques nouvelles. J'aimerai simplement évoquer quelques voies qui me semble importantes dans ce contexte :

– Nous avons recherché des méthodes permettant de pallier l'absence de planification résultant du calcul en ligne de la commande. Il est possible, grâce au formalisme de redondance de l'approche fonction de tâche, de définir une solution à ce problème. L'approche originale que nous avons proposée reposant sur la résolution d'un

système d'équations linéaires doit pouvoir être étendue et considérée dans d'autres contextes comme l'évitement d'obstacles (où les contraintes pourraient s'exprimer sous la forme d'inégalités sur les positions articulaires ou même sur la position de l'effecteur dans l'image dans le cas d'un asservissement visuel déporté), la limitation des vitesses articulaires (inégalité sur les vitesses articulaires) ou des accélérations articulaires, etc. De la même manière, le cas des contraintes exprimées dans l'image pourrait sans doute tirer partie d'une telle approche. Ces différents objectifs nécessiteront évidemment des modifications importantes de l'approche déjà définie mais ouvrent des axes de recherche et des champs applicatifs variés.

– Les études menées au sein des projets Vista puis, désormais, dans l'équipe lagad IC dans le domaine de l'asservissement visuel et de la vision active concernaient essentiellement l'utilisation d'une seule caméra. Dans un contexte de scènes plus complexes, encombrées ou comportant des objets mobiles, nous souhaitons à présent généraliser cette approche à l'emploi de deux capteurs, l'un fournissant une vision locale embarquée et l'autre ayant une vue globale de la scène. Nous avions démontré et validé, dans le cas simple d'un manipulateur à trois degrés de liberté, une tâche d'insertion (*peg in hole*) *[139]* dans un environnement encombré. L'utilisation d'un second capteur permettant d'obtenir une vision globale de la scène a été introduite permettant de caractériser les obstacles et de planifier partiellement la trajectoire de la caméra. Cette planification avait été réalisée en utilisant des fonctions de navigation introduites dans la commande en utilisant la redondance. La généralisation à un manipulateur à n degrés de liberté n'a cependant pas été réalisée et reste un problème à considérer dans le futur. D'autres part, nous avons pour le moment considéré que la seconde caméra donnant une vue globale de la scène était fixe. Cette hypothèse devra être levée pour pouvoir appréhender des problèmes réels. Sa trajectoire devra être commandée pour maintenir l'effecteur du manipulateur dans le champ de vision [176].
– La loi de commande robuste d'asservissement visuel que nous avons proposée donne le plus souvent de très bon résultats et permet de rejeter pour un coût calculatoire relativement faible la plupart des données aberrantes. Il existe cependant quelques cas particuliers où cet algorithme est mis en échec. Ce problème est dû, de manière générale, à la minimisation robuste de fonctions multivariables et dans notre cas particulier à la difficulté de différencier les erreurs dans l'image dues au mouvement sur chaque degrés de liberté de la caméra. Ce problème difficile à résoudre se retrouve bien évidemment dans un contexte d'asservissement visuel, mais aussi pour ne citer que les problèmes qui nous ont intéressé dans le cadre de nos travaux, celui du suivi par calcul de pose non linéaire.
– Dans un tout ordre d'idée et concernant l'étude menée dans le contexte de l'animation sur le positionnement par rapport à des sources lumineuses (section 4.2 et *[138]*), il nous semble intéressant de valider ces résultats sur le site robotique du projet. Dans un contexte de vision robotique, assurer un bon éclairage de la scène, en modifiant la position des sources lumineuses est en effet un problème crucial en particulier pour faciliter le traitement d'images. Des premières expériences dans ce sens ont récemment été réalisées et devront être poursuivies.

A plus long terme, nous souhaitons lancer de nouvelles recherches sur les aspects de modélisation d'informations visuelles en considérant directement les niveaux de gris de l'image, ou d'une zone de l'image. Cela permettrait de réduire à sa plus simple expression la phase de traitement d'images, toujours problématique et source potentielle d'erreur. La phase de modélisation s'avèrera sûrement difficile, les informations n'étant plus seulement de nature géométrique mais aussi photométrique. S'il est possible d'établir une matrice Jacobienne reliant la variation de l'intensité lumineuse au déplacement 2D dans l'image [75, 91], aboutir à des résultats similaires mais pour des déplacement 3D du capteur semble beaucoup plus difficile au vu des nombreux paramètres entrant en ligne de compte. Les nombreux travaux réalisés sur l'analyse du mouvement 3D semblent cependant une piste à examiner. Une autre piste à suivre est issue des travaux que nous avons réalisés, sur le positionnement par rapport à des sources lumineuses qui pourrait être étendus à des taches de positionnement ou de poursuite.

Animation Dans le cadre de la thèse de Nicolas Courty, nous avons débuté une étude portant sur les liens entre l'animation et l'asservissement visuel. Dans le domaine de l'animation, le mouvement des caméras est en effet souvent décrit dans l'espace 3D, ce qui rend difficile la spécification de telles trajectoires. Les techniques d'asservissement visuel permettent, elles, de spécifier ces trajectoires en fonction des informations perçues dans l'image. Cela permet d'une part une description de la tâche à réaliser plus "intuitive" et d'autre part d'être à même de prendre en compte automatiquement et en temps réel des modifications de l'environnement. En mettant à profit notre savoir-faire dans le domaine, l'asservissement visuel est apparu comme une solution efficace pour gérer le problème du déplacement d'entités virtuelles (caméras *[138]* ou humanoïdes *[46]*) dans des mondes virtuels et l'animateur peut donc disposer de nouvelles fonctionnalités importantes dans son système d'animation.

Cette étude pourrait se poursuivre à l'avenir dans plusieurs directions.

- Nous avons vu que l'absence de contrôle de la trajectoire de la caméra dû à l'utilisation d'un asservissement visuel 2D, pouvait être considéré comme un défaut de l'approche proposée. Il conviendra donc d'étudier comment résoudre ce problème, en recourant par exemple à des loi de commande 2D 1/2 ou 3D, tout en essayant de préservant au maximum la possibilité spécifier les tâches dans un espace 2D.
- Une des difficultés de l'approche proposée réside dans la coexistence possible, pour une même tâche, de différentes fonctions de coût à minimiser. La simple combinaison linéaire des tâches secondaires n'est pas une solution satisfaisante à ce problème et la hiérarchisation des tâches afin de pouvoir effectuer des projections en cascade [10] est une solution beaucoup plus élégante à ce problème. Il reste que dans le cadre de l'animation et afin de faciliter le travail de l'animateur il serait souhaitable de pouvoir réaliser cette décomposition de façon automatique.
- Concernant plus particulièrement l'animation de chaînes articulées (tels les humanoïdes), les solutions que nous avons proposées à ce jour pour la gestion de grands déplacements (c'est-à-dire principalement sur la marche) ne sont pas totalement satisfaisantes. En effet, dans le module de locomotion par asservissement visuel, les effets de la locomotion sur la perception n'ont pas été considérés, ce qui entraîne des perturbations significatives dans le système. La simple résolution de la tâche visuelle

ne suffit pas pour garantir le réalisme du mouvement produit (il faudrait pouvoir tenir compte de la nature dynamique du mouvement, en considérant l'inertie de l'humanoïde ou bien encore des problèmes d'équilibre).
- Par ailleurs, il pourrait aussi être intéressant de considérer la commande des membres supérieurs (bras, mains) de l'humanoïde via un processus d'asservissement visuel. Dans cette optique, les recherches que nous avons réalisées en collaboration avec Ifremer sur l'utilisation d'une caméra déportée, dans un tout autre contexte *[136]*, pourraient s'avérer utiles.
- Nos récents travaux sur la simulation de la perception visuelle *[45]* semblent constituer une piste de recherche intéressante d'une part dans un contexte de simulation et d'autre part dans un contexte de vidéo surveillance pour le choix automatique des éléments visuels à observer. Le modèle théorique de perception reste à améliorer (par exemple par ajout d'autres types d'informations visuelles [88] et en proposant d'autres méthodes pour la fusion des différentes cartes de saillance).
- Enfin, l'un des objectifs de la thèse de Nicolas Courty a été de définir des stratégies d'action des avatars en fonction de leur perception afin d'accroître leur autonomie d'évolution et leur degré de crédibilité. Pour cela, les objets composant un environnement doivent offrir aux avatars des informations sur la façon de se comporter pour réaliser avec/sur eux telle ou telle tâche (principe de "*affordances*").

Suivi d'objets et localisation 3D. Le suivi d'objets ou de formes dans une séquence d'images est un problème clé, aussi bien en tant que sujet de recherches en vision par ordinateur, que pour transférer nos recherches en asservissement visuel. Les techniques d'asservissement visuel étant à la base des techniques de commande "bas niveau", il faut les intégrer dans des systèmes de plus haut niveau pour en assurer une véritable diffusion. *A priori*, ce travail important concerne la conception d'IHM, l'introduction de systèmes de reconnaissance d'objets, ou encore la définition puis éventuellement la robustification des algorithmes de traitements d'images. En pratique notre contribution s'est située essentiellement au niveau de la définition d'algorithmes de suivi d'objets rapides et robustes dans des séquences d'images.

Nous poursuivons actuellement nos travaux sur ce point dans le cadre de la thèse d'Andrew Comport. La méthode de suivi et de localisation 3D que nous avons proposée ("asservissement visuel virtuel") pouvant être considérée comme une approche duale de l'asservissement visuel "classique", tout le savoir-faire en asservissement visuel peut s'appliquer assez directement à cette problématique classique en vision par ordinateur. Afin de prendre en compte des données aberrantes, nous avons introduit dans les lois de commande des estimateurs robustes (M-estimateurs). Cette approche statistique permet de détecter des erreurs dans l'extraction des données ou leur mise en correspondance et d'obtenir une estimation beaucoup plus précise de la pose. Un travail important a aussi été réalisé sur la modélisation des informations visuelles utilisées dans la loi de commande. Les algorithmes résultants permettent un suivi 3D à la cadence vidéo *[36]* d'objets complexes dont le modèle CAO est connu.

Dans un avenir proche, nous étendrons ces travaux dans différentes directions. Sur les aspects de suivi en temps réel, les travaux et perspectives décrits dans le paragraphe précédent considèrent un suivi principalement 3D nécessitant la connaissance (parfois trop forte

pour bon nombre d'applications) du modèle 3D des objets considérés. Dans le cadre de la thèse de Muriel Pressigout (qui a débuté en septembre 2003) nous souhaitons développer des algorithmes de suivi 2D reposant à la fois sur des informations de type contour (zones de fort gradient) et des informations de type texture (zones où il est plus fiable et aisé d'estimer le mouvement 2D). Nous comptons modéliser les contours de l'objet, de même que son mouvement, par des modèles paramétriques dont il faut estimer les caractéristiques (modèle projectif de mouvement, déformations, etc.). Il faudra aussi prendre en compte des informations visuelles de types différents et développer des algorithmes robustes aux mesures aberrantes et à de potentielles occultations. Les algorithmes ainsi développés devront s'exécuter à la cadence vidéo pour pouvoir être intégrés dans des applications de vision robotique.

Concernant plus directement le suivi 3D, des améliorations devraient pouvoir être apportées en considérant deux pistes distinctes (mais complémentaires).

- La première consiste à introduire dans le suivi une contrainte spatio-temporelle. Dans l'algorithme actuel, chaque nouvelle image est considérée indépendamment des précédentes (si ce n'est pour l'initialisation du suivi). La géométrie d'un système multi-caméras ou d'une caméra en mouvement induit cependant de fortes contraintes sur la mise en correspondance. L'estimation conjointe de la pose et du déplacement de la caméra devrait donc permettre d'obtenir une meilleure précision et une meilleure stabilité de la localisation 3D. Dans le cadre de la thèse de Muriel Pressigout, nous pensons pouvoir estimer conjointement la pose et le déplacement (représenté sous la forme d'une homographie) en couplant à la fois l'apparence [75, 91] et les contours *[36]*[117, 60]. Considérer conjointement la pose et le déplacement de la caméra dans un unique processus d'optimisation introduira implicitement la contrainte temporelle qui manque dans nos travaux actuels.
- Une deuxième direction à explorer est le couplage multi-capteurs (caméra et localisateur magnétique ou inertiel). La fusion des informations provenant de ces différents capteurs grâce à un filtrage de type Kalman (ou autre), après une phase de calibration, devrait permettre assez facilement de prendre en compte des mouvements beaucoup plus rapides de la caméra ou des objets suivis.

Une autre piste consiste à considérer d'autres modèles de projection que la projection perspective classique, par exemple les modèles de projection correspondant aux capteurs de vision omnidirectionnelle, que nous avons commencés à étudier dans le cadre du projet Robea Omnibot.

Finalement, nous avons pour objectif à plus long terme d'étendre ces travaux aux cas d'objets non seulement articulés mais déformables. Dans ce cas, il s'agit d'estimer non seulement la localisation 3D, mais aussi les déformations que subit l'objet. Une autre évolution serait de considérer la reconstruction et le suivi d'objets ou de scènes 3D représentées par des graphes de scène paramétrés [54] et/ou de modèles topologiques paramétrés.

Enfin produire un système informatique permettant à moindre coût et dans des délais réduits de prototyper tant des systèmes de vision robotique, des système d'animation ou encore des systèmes de réalité augmentée a toujours été l'un de nos objectifs. Après avoir expérimenté dans un premier temps les langages synchrones *[171, 140]* nous avons fina-

lement retenu un langage plus “classique” pour réaliser cette plate-forme. ViSP *[126]* est écrit en C++ et utilise largement les fonctionnalités des langages orientés objets, en particulier pour assurer son évolution future et sa portabilité. Tous les travaux décrits et postérieurs à 1998 reposent sur son utilisation. Par ailleurs l’ensemble des développements logiciels de l’équipe lagadic sont construits à partir de ViSP, qui les intègre au fur et à mesure. Nos développements en réalité augmentée sont également réalisés à l’aide de cet environnement logiciel.

Épilogue

Nous sommes donc partis d'une problématique visant à commander une caméra bien réelle montée sur l'effecteur d'un robot à partir d'informations extraites d'images du monde réel. Cette problématique a ensuite évoluée afin de commander les mouvements d'une caméra virtuelle dans un monde virtuel. Finalement afin de répondre aux besoins récurrents en traitement de séquences vidéo, nous avons proposé une approche de suivi reposant sur la commande d'une caméra virtuelle dans un monde réel... Ce dernier algorithme est finalement utilisé pour commander la caméra réelle initialement considérée. J'espère donc avoir convaincu le lecteur que la boucle perception-action sur laquelle repose l'ensemble de nos travaux est complètement, mais pas définitivement, fermée.

ANNEXE A

Quelques notations

Notations générales

- a, λ : scalaire noté en minuscule
- $\mathbf{v}$: vecteur noté en minuscule gras
- $\mathbf{M}$: matrice notée en majuscule gras
  - $\mathbf{M}_{ij}$ désigne l'élément situé à l'intersection de la i-ème ligne et de la j-ème colonne de la matrice
  - $\mathbf{M}_{i\bullet}$ désigne la i-ème ligne de la matrice
  - $\mathbf{M}_{\bullet j}$ désigne la j-ème colonne de la matrice
- $[\mathbf{v}]_\times$ matrice de préproduit vectoriel associée au vecteur $\mathbf{v}$

Changements de repère – transformations

- ${}^a\mathbf{M}_b$: matrice homogène définie par

$$ {}^a\mathbf{M}_b = \begin{pmatrix} {}^a\mathbf{R}_b & {}^a\mathbf{t}_b \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} $$

  où ${}^a\mathbf{R}_b$ est une matrice de rotation et ${}^a\mathbf{t}_b$ un vecteur de translation. ${}^a\mathbf{M}_b$ décrit la position du repère a dans le repère b.
- ${}^c\mathbf{M}_o$: Cas particulier de la matrice de pose : ${}^c\mathbf{M}_o$ désigne la position de la caméra c dans le repère de la scène o.
- ${}^a\mathbb{T}_b$ est la matrice de changement de repère pour le torseur cinématique. ${}^a\mathbb{T}_b$ est une matrice 6×6 donnée par :

$$ {}^a\mathbb{T}_b = \begin{pmatrix} {}^a\mathbf{R}_b & [{}^a\mathbf{t}_b]_\times {}^a\mathbf{R}_b \\ \mathbf{0}_{3\times 3} & {}^a\mathbf{R}_b \end{pmatrix} $$

Asservissement visuel

- $\mathbf{v}$: vitesse de la caméra (torseur cinématique). $\mathbf{v} = (\mathbf{v}, \omega)$ où :
  - $\mathbf{v} = (v_x, v_y, v_z)$: vitesse de translation
  - $\omega = (\omega_x, \omega_y, \omega_z)$: vitesse de rotation
- $\mathbf{s}$: primitives visuelles (signaux capteur)
- $\mathbf{s}^*$: consigne (position désirée de $\mathbf{s}$)
- $\mathbf{L_s}$: matrice d'interaction relative à $\mathbf{s}$
- $\widehat{\mathbf{L_s}}$: modèle ou estimation de la matrice d'interaction relative à $\mathbf{s}$
- $\mathbf{H_s}$: matrice d'interaction dans l'espace articulaire

Robotique

- $\mathbf{q}$: vecteur des coordonnées articulaires du manipulateur
- $\bar{\mathbf{q}}_{min}$ et $\bar{\mathbf{q}}_{max}$: butées basses et hautes du manipulateur
- $\mathbf{J}(\mathbf{q})$: Jacobien du robot.

Bibliographie

[1] Maciejewski A.A., C.A. Klein. Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *Int. Journal of Robotics Research*, 4(3):109–117, Fall 1985.

[2] Y. Aloimonos. Purposive and qualitative active vision. In *IAPR Int. Conf. on Pattern Recognition, ICPR'90*, volume 1, pp. 346–360, Atlantic City, New Jersey, juin 1990.

[3] Y. Aloimonos, I. Weiss, A. Bandopadhay. Active Vision. *Int. Journal of Computer Vision*, 1(4):333–356, janvier 1987.

[4] N. Andreff, B. Espiau, R. Horaud. Visual servoing from lines. *Int. Journal of Robotics Research*, 21(8):769–700, août 2002.

[5] D. Arijon. *Grammar of the Film Language*. Communication Arts Books, Hastings House, New York, 1976.

[6] I. Asimov. *I, Robot*. Gnome Press, 1950.

[7] R. Azuma. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, août 1997.

[8] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, B. MacIntyre. Recent Advances in Augmented Reality. *IEEE Computer Graphics and Application*, 21(6):34–47, novembre 2001.

[9] N. Badler, M. Palmer, R. Bindiganavale. Animation control for real-time virtual humans. *Communications of the ACM*, 42(8):64–73, août 1999.

[10] P. Baerlocher, R. Boulic. Task-priority Formulations for the Kinematic Control of Highly Redundant Articulated Structures. In *IROS'98*, Victoria, Canada, octobre 1998.

[11] P. Baerlocher, R. Boulic. An Inverse Kinematic Architecture Enforcing an Arbitrary Number of Strict Priority Levels. *The Visual Computer*, 2004.

[12] R. Bajcsy. Active Perception. *Proceedings of the IEEE*, 76(8):996–1005, août 1988.

[13] D.H. Ballard. Animate vision. *Artificial Intelligence*, 48(1):57–86, février 1991.

[14] M.-O. Berger. How to track efficiently piecewise curved contours with a view to reconstructing 3D objects. In *Int. Conf on Pattern Recognition, ICPR'94*, pp. 32–36, Jerusalem, octobre 1994.

[15] A. Blake, M. Isard. *Active Contours*. Springer Verlag, avril 1998.

[16] J. Blinn. Where Am I ? What Am I Looking At ? *IEEE Computer Graphics and Application*, pp. 76–81, juillet 1998.

[17] S. Boukir. *Reconstruction 3D d'un environnement statique par vision active*. Thèse de doctorat, Université de Rennes 1, IRISA, octobre 1993.

[18] S. Boukir, P. Bouthemy, F. Chaumette, D. Juvin. A local method for contour matching and its parallel implementation. *Machine Vision and Application*, 10(5/6):321–330, avril 1998.

[19] R. Boulic, D. Mas, R. Thalmann. Complex Character Positioning Based on a Compatible Flow Model of Multiple Supports. *IEEE Trans. on Visualization and Computer Graphics*, 3(3):245 – 261, juillet 1997.

[20] P. Bouthemy. A Maximum Likelihood Framework for Determining Moving Edges. *IEEE Trans. on Pattern Analysis and Machine intelligence*, 11(5):499–511, mai 1989.

[21] E. Boyer, J.-S. Franco. A Hybrid Approach for Computing Visual Hulls of Complex Object. In *IEEE, Int. Conf. on Computer Vision and Pattern Recognition, CVPR'03*, volume 1, pp. 695–701, 2003.

[22] P. Braud, M. Dhome, J.-T. Lapresté, B. Peuchot. Reconnaissance, localisation et suivi d'objets polyhédriques par vision multi-oculaire. *Technique et Science Informatiques*, 16(1):9–38, Janvier 1997.

[23] D.C. Brown. Close-Range Camera Calibration. *Photogrammetric Engineering*, 4(2):127–140, mars 1971.

[24] V. Cadenat, R. Swain, P. Soueres, M. Devy. A controller to perform a visually guided tracking task in a cluttered environment. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'99*, pp. 775–780, Kyongju, Korea, Octobre 1999.

[25] B. Capre, V. Torre. Using vanishing points for camera calibration. *Int. J. of computer Vision*, pp. 127–140, 1990.

[26] T.-F. Chang, R.-V. Dubey. A Weighted Least-Norm Solution Based Scheme for Avoiding Joints Limits for Redundant Manipulators. *IEEE Trans. on Robotics and Automation*, 11(2):286–292, avril 1995.

[27] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In D.J. Kriegman, G. Hager, A.S. Morse, Eds., *The confluence of vision and control*, Lecture Notes in control and information sciences, No 237, pp. 67–78. Springer, juin 1997.

[28] F. Chaumette. Asservissement visuel. In W. Khalil, Ed., *La commande des robots manipulateurs*, Traité IC2, chapter 3, pp. 105–150. Hermès, 2002.

[29] F. Chaumette. Avancées récentes en asservissement visuel. In *Journées Nationales de la Recherche en Robotique, JNRR'03*, pp. 103–108, Clermont-Ferrand, October 2003.

[30] F. Chaumette. Image moments: a general and useful set of features for visual servoing. *IEEE Trans. on Robotics*, 20(4), août 2004.

[31] F. Chaumette, S. Boukir, P. Bouthemy, D. Juvin. Structure from controlled motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(5):492–504, mai 1996.

[32] F. Chaumette, E. Marchand. A redundancy-based iterative scheme for avoiding joint limits: Application to visual servoing. *IEEE Trans. on Robotics and Automation*, 17(5):719–730, Octobre 2001.

[33] F. Chaumette, P. Rives. Modélisation et calibration d'une caméra. In *7ème congrès AFCET Reconnaissance des formes et intelligence artificielle, RFIA'89*, volume 1, pp. 527–536, Paris, décembre 1989.

[34] S. Christy, R. Horaud. Iterative Pose Computation from Line Correspondences. *Computer Vision and Image Understanding*, 73(1):137–144, janvier 1999.

[35] C. Colombo, B. Allotta, P. Dario. Affine visual servoing: a framework for relative positioning with a robot. In *IEEE Int. Conference on Robotics and Automation ICRA'95*, pp. 464–471, Nagoya, Japan, mai 1995.

[36] A. Comport, E. Marchand, F. Chaumette. A real-time tracker for markerless augmented reality. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03*, pp. 36–45, Tokyo, Japan, octobre 2003.

[37] A. Comport, E. Marchand, F. Chaumette. Object-based visual 3D tracking of articulated objects via kinematic sets. In *IEEE Workshop on Articulated and Non-Rigid Motion*, Washington, DC, June 2004.

[38] A. Comport, E. Marchand, F. Chaumette. Robust model-based tracking for robot vision. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'04*, Sendai, Japan, september 2004.

[39] A. Comport, M. Pressigout, E. Marchand, F. Chaumette. A Visual Servoing Control Law that is Robust to Image Outliers. In *IEEE Int. Conf. on Intelligent Robots and Systems, IROS'03*, volume 1, pp. 492–497, Las Vegas, Nevada, octobre 2003.

[40] C. Connolly. The Determination of Next Best Views. In *IEEE Int. Conf. on Robotics and Automation*, pp. 432–435, St Louis, Missouri, mars 1985.

[41] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham. Active shape models - their training and application. *CVGIP : Image Understanding*, 61(1):38–59, Janvier 1994.

[42] E. Coste-Manière, P. Couvignou, P. Kohsla. Robotic Contour Following based on Visual Servoing. In *IEEE/RSJ, Int. Conference on Intelligent Robots and Systems IROS'93*, Yokohama, Japan, juillet 1993.

[43] N. Courty. *Animation référencée vision : de la tâche au comportement*. Thèse de doctorat, INSA de Rennes, novembre 2002.

[44] N. Courty, F. Lamarche, S. Donikian, E. Marchand. A Cinematography System for Virtual Storytelling. In O. Balet, G. Subsol, P. Torguet, Eds., *Int. Conf. on Virtual Storytelling, ICVS'03*, volume 2897 of *Lecture Notes in Computer Science*, pp. 30–34, Toulouse, France, novembre 2003.

[45] N. Courty, E. Marchand. Visual perception based on salient features. In *IEEE Int. Conf. on Intelligent Robots and Systems, IROS'03*, volume 2, pp. 1024–1029, Las Vegas, Nevada, Octobre 2003.

[46] N. Courty, E. Marchand, B. Arnaldi. Through-the-eyes control of a virtual humanoïd. In H.-S. Ko, Ed., *IEEE Computer Animation, CA'01*, pp. 74–83, Seoul, South Korea, novembre 2001.

[47] N. Courty, E. Marchand, B. Arnaldi. A new application for salliency maps: Synthetic vision of autonomous actors. In *IEEE Int. Conf. on Image Processing, ICIP'03*, volume 3, pp. 1065–1068, Barcelona, Spain, Septembre 2003.

[48] C.K. Cowan, P.D. Kovesi. Automatic Sensor Placement from Vision task Requirements. *IEEE Trans. on Pattern Analysis and Machine intelligence*, 10(3):407–416, mai 1988.

[49] N.J. Cowan, J.D. Weingarten, D.E. Koditschek. Visual Servoing Via Navigation Functions. *IEEE Trans. on Robotics and Automation*, 18(4):521– 533, août 2002.

[50] J. Crowley, M. Mesrabi, F. Chaumette. Comparison of kinematic and visual servoing for fixation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'95*, volume 1, pp. 335–341, Pittsburgh, Pennsylvania, August 1995.

[51] A. Crétual, F. Chaumette. Visual servoing based on image motion. *Int. Journal of Robotics Research*, 20(11):857–877, novembre 2001.

[52] N. Daucher, M. Dhome, J.T. Lapreste, G. Rives. Modelled Object Pose Estimation and Tracking by Monocular Vision. In *British Machine Vision Conference, BMVC'93*, pp. 249–258, Guildford, UK, septembre 1993.

[53] S. de Ma. Conics-based stereo, motion estimation and pose determination. *Int. Journal of Computer Vision*, 10(1):7–25, 1993.

[54] P. Debevec, C.-J. Taylor, J. Malik. Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, pp. 11–20, New Orleans, Louisiana, août 1996.

[55] K. Deguchi. A Direct Interpretation of Dynamic Images with Camera and Object Motions for Vision Guided Robot Control. *Int. Journal of Computer Vision*, 37(1):7–20, juin 2000.

[56] D. Dementhon, L. Davis. Model-Based Object Pose in 25 Lines of Codes. *Int. J. of Computer Vision*, 15:123–141, 1995.

[57] M. Dhome, J.-T. Lapresté, G. Rives, M. Richetin. Determination of the attitude of modelled objects of revolution in monocular perspective vision. In *European Conference on Computer Vision, ECCV'90*, volume 427 of *Lecture Notes in Computer Science*, pp. 475–485, Antibes, avril 1990.

[58] M. Dhome, M. Richetin, J.-T. Lapresté, G. Rives. Determination of the Attitude of 3D Objects from a Single Perspective View. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(12):1265–1278, décembre 1989.

[59] S.M. Drucker, D. Zeltzer. Intelligent Camera Control in a Virtual Environment. In *Graphics Interface'94*, pp. 190–199, Banff, Canada, 1994.

[60] T. Drummond, R. Cipolla. Real-Time Visual Tracking of Complex Structures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(7):932–946, juillet 2002.

[61] B. Espiau, R. Boulic. Collision Avoidance for Redundant Robots with Proximity Sensors. In *Int. Symposium of Robotic Research*, Gouvieux, France, Octobre 1985.

[62] B. Espiau, F. Chaumette, P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, juin 1992.

[63] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.

[64] O.D Faugeras, G. Toscani. Camera calibration for 3D computer vision. In *Proc Int. Workshop on Machine Vision and Machine Intelligence*, pp. 240–247, Tokyo, février 1987.

[65] N. Fischler, R.C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography. *Communication of the ACM*, 24(6):381–395, juin 1981.

[66] G. Flandin, F. Chaumette, E. Marchand. Eye-in-hand / Eye-to-hand Cooperation for Visual Servoing. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, pp. 2741–2746, San Francisco, CA, avril 2000.

[67] S. Ganapathy. Decomposition of Transformation Matrices for Robot Vision. *Pattern Recognition Letter*, 2:401–412, 1984.

[68] J. Gangloff, M. de Mathelin, G. Abba. 6 DOF High Speed Dynamic Visual Servoing Using GPC Controllers. In *IEEE Int. Conf. on Robotics and Automation, ICRA'98*, pp. 2008–2013, mai 1998.

[69] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar, N. Navab. Marker-less Tracking for AR: A Learning-Based Approach. In *IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR'02)*, pp. 3–6, Darmstadt, Germany, septembre 2002.

[70] D.B. Gennery. Visual tracking of known three-dimensional objects. *Int. J. of Computer Vision*, 7(3):243–270, 1992.

[71] M. Girard, A.A. Maciejewski. Computational modeling for the computer animation of legged figures,. *ACM SIGGRAPH conference, Computer Graphics*, 19(3):263–270, juillet 1985.

[72] M. Gleicher, A. Witkin. Through-the-lens camera control. In *ACM Computer Graphics, SIGGRAPH'92*, pp. 331–340, Chicago, juillet 1992.

[73] E. Grosso, G. Metta, A. Oddera, G. Sandini. Robust visual servoing in 3D reaching tasks. *IEEE Trans. on Robotics and Automation*, 12(5):732–742, octobre 1996.

[74] G. Hager, P. Belhumeur. Efficient Region Tracking With Parametric Models of Geometry and Illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, octobre 1998.

[75] G. Hager, K. Toyama. The XVision System: A General-Purpose Substrate for Portable Real-Time Vision Applications. *Computer Vision and Image Understanding*, 69(1):23–37, janvier 1998.

[76] R. Haralick, H. Joo, C. Lee, X. Zhuang, V Vaidya, M. Kim. Pose estimation from corresponding point data. *IEEE Trans on Systems, Man and Cybernetics*, 19(6):1426–1445, novembre 1989.

[77] R. Hartley, A. Zisserman. *Multiple View Geometry in computer vision*. Cambridge University Press, 2001.

[78] K. Hashimoto, Ed. *Visual Servoing : Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific Series in Robotics and Automated Systems, Vol 7, World Scientific Press, Singapor, 1993.

[79] K. Hashimoto, H. Kimura. Dynamic Visual Servoing With Nonlinear Model-Based Control. In *Proceedings of the 12th World Congress IFAC*, volume 9, pp. 405–408, Sidney, Autralia, juillet 1993.

[80] K. Hashimoto, H. Kimura. LQ Optimal and non-linear approaches to visual servoing. In K. Hashimoto, Ed., *Visual Servoing*, volume 7, pp. 165–198. World Scientific Series in Robotics and Automated Systems, Singapour, 1993.

[81] L.-W. He, M.F. Cohen, D.H. Salesin. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In *ACM SIGGRAPH'96, in Computer Graphics Proceedings*, pp. 217–224, New Orleans, août 1996.

[82] G. Hégron, B. Arnaldi, C. Lecerf. *Computer Animation*. Prentice Hall, Juillet 1995.

[83] R. Horaud, B. Conio, O. Leboulleux, B. Lacolle. An analytic solution for the perspective 4-points problem. *Computer Vision, Graphics and Image Processing*, 47(1):33–44, juillet 1989.

[84] B. Horn. *Robot Vision*. MIT Press, Cambridge, 1987.

[85] K. Hosoda, M. Asada. Versatile visual servoing without knowledge of true jacobian. In *IEEE/RSJ Int. Conf on Intelligent Robots and Systems, IROS'94*, pp. 186–193, Munich, Germany, août 1994.

[86] P.-J. Huber. *Robust Statistics*. Wiler, New York, 1981.

[87] S. Hutchinson, G. Hager, P. Corke. A tutorial on Visual Servo Control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, octobre 1996.

[88] L. Itti, C. Koch, E. Niebur. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, Nov 1998.

[89] M. Jägersand, O. Fuentes, R. Nelson. Experimental evaluation of uncalibrated visual servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'97*, volume 3, pp. 2874–2880, Albuquerque, NM, avril 1997.

[90] A.K. Jain, Y. Zhong, S. Lakshmanan. Object matching using deformable templates. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 18(3):267–278, mars 1996.

[91] F. Jurie, M. Dhome. Hyperplane Approximation for Template Matching. *IEEE trans on Pattern Analysis and Machine Intelligence*, 24(7):996–1000, juillet 2002.

[92] M. Kass, A. Witkin, D. Terzopolous. Snakes : Active contour models. In *Int. Conf. Computer Vision, ICCV'87*, pp. 259–268, London, UK, 1987.

[93] H. Kato, M. Billinghurst. Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System. In *ACM/IEEE Int. Workshop on Augmented Reality, IWAR'99*, pp. 85–95, San Francisco, CA, octobre 1999.

[94] C. Kervrann, F. Heitz. A Hierarchical Markov Modeling Approach for the Segmentation and Tracking of Deformable Shapes. *Graphical Models and Image Processing*, 60(3):173–195, mai 1998.

[95] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. Journal of Robotics Research*, 5(1):90–98, 1986.

[96] K. Kinoshita, K. Deguchi. Simultaneous Determination of Camera Pose and Intrinsic Parameters by Visual Servoing. In *IAPR Int. Conf. on Pattern Recognition, ICPR'94*, volume A, pp. 285–289, Jerusalem, 1994.

[97] G. Klein, T. Drummond. Robust Visual Tracking for Non-Instrumented Augmented Reality. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03*, pp. 113–122, Tokyo, Japan, octobre 2003.

[98] D. Koller, K. Daniilidis, H.-H. Nagel. Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes. *Int. Journal of Computer Vision*, 10(2):257–281, juin 1993.

[99] D. Koller, G. Klinger, E. Rosse, D. Breen, R. Whitaker, M. Tuceryan. Real-time Vision-Based Camera Tracking for augmented reality applications. In *Int. Symp. on Virtual Reality Software and Technology, VRST'97*, pp. 87–94, Lausanne, Switzerland, septembre 1997.

[100] H. Kollnig, H.-H. Nagel. 3D Pose Estimation by fitting Image Gradients Directly to Polyhedral Models. In *IEEE Int. Conf. on Computer Vision*, pp. 569–574, Boston, MA, mai 1995.

[101] D. Kragic, H. Christensen. Cue integration for visual servoing. *IEEE Trans. on Robotics and Automation*, 17(1):19–26, février 2001.

[102] D. Kragic, H.I. Christensen. Model Based Techniques for Robotic Servoing and Grasping. In *IEEE Int. Conf. on intelligent robots and systems, IROS'02*, volume 1, pp. 299–304, Lausanne, Switzerland, octobre 2002.

[103] D. Kragic, H.I. Christensen. Confluence of Parameters in Model Based Tracking. In *IEEE Int. Conf. on Robotics and Automation, ICRA'03*, volume 4, pp. 3485–3490, Taipe, Taiwan, septembre 2003.

[104] A. Krupa, J. Gangloff, C. Doignon, M. de Mathelin, G. Morel, J. Leroy, L. Soler, J. Marescaux. Autonomous 3D positioning of surgical instruments in robotized laparoscopic surgery using visual servoing. *IEEE Trans. on robotics and automation*, 19(5):842 –853, octobre 2003.

[105] J. Kuffner. *Autonoumous agents for real-time animation*. Thèse de doctorat, Stanford university, décembre 1999.

[106] R. Kumar, A.R. Hanson. Robust methods for estimating pose and a sensitivity analysis. *CVGIP: Image Understanding*, 60(3):313–342, novembre 1994.

[107] K.N. Kutulakos, C.R. Dyer. Globarface Reconstruction by Purposive Control of Observer Motion. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 339–346, Seattle, USA, juin 1994.

[108] K.N. Kutulakos, S.M. Seitz. A Theory of Shape by Space Carving. *International Journal of Computer Vision*, 38(3):199–218, juillet 2000.

[109] M.H. Kyung, M.-S. Kim, S. Hong. Through-the-Lens Camera Control with a Simple Jacobian Matrix. In *Graphics Interface'95*, pp. 171–178, Quebec, Canada, mai 1995.

[110] J.-T. Lapresté, F. Jurie, M. Dhome, F. Chaumette. An Efficient Method to Compute the Inverse Jacobian Matrix in Visual Servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, New Orleans, avril 2004.

[111] J.-M. Lavest, M. Viala, M. Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration ? In *European Conference on Computer Vision, ECCV'98*, volume 1, pp. 158–174, Freiburg, Germany, juin 1998.

[112] J. Lee, S. Shin. A Hierarchical Approach to Interactive Motion Editing for Human-Like Figures. In Alyn Rockwood, Ed., *SIGGRAPH 99, in Computer Graphics Proceedings*, pp. 39–48, New-york, USA, août 1999.

[113] V. Lepetit, L. Vacchetti, T. Thalmann, P. Fua. Fully Automated and Stable Registration for Augmented Reality Applications. In *ACM/IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03*, pp. 93–102, Tokyo, Japan, octobre 2003.

[114] A. Liegeois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Trans. on Systems, Man and Cybernetics*, 7(12):868–871, décembre 1977.

[115] Y. Liu, T.S. Huang, O.D. Faugeras. Determination of Camera Location from 2-D to 3D Line and Point Correspondences. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 12(1):28–37, janvier 1990.

[116] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–394, mars 1987.

[117] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, mai 1991.

[118] C.P. Lu, G.D. Hager, E. Mjolsness. Fast and Globally Convergent Pose Estimation from Video Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(6):610–622, juin 2000.

[119] B.D. Lucas, T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Int. Joint Conf. on Artificial Intelligence, IJCAI'81*, pp. 674–679, 1981.

[120] N. Magnenat-Thalmann, D. Thalmann. Virtual Actors living in a Real World. In *IEEE Computer Animation, CA'95*, pp. 19–29, Geneva, Switzerland, avril 1995.

[121] E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, volume 2, pp. 1843–1848, New Orleans, avril 2004.

[122] E. Malis, F. Chaumette, S. Boudet. 2 1/2 D Visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):238–250, avril 1999.

[123] E. Malis, F. Chaumette, S. Boudet. 2 1/2 D Visual Servoing with Respect to Unknown Objects Through a New Estimation Scheme of Camera Displacement. *Int. Journal of Computer Vision*, 37(1):79–97, juin 2000.

[124] E. Malis, P. Rives. Robustness of Image-Based Visual Servoing with Respect to Depth Distribution Errors. In *IEEE International Conference on Robotics and Automation*, volume 2, pp. 1056–1061, Taipei, Taiwan, septembre 2003.

[125] S. Marcelja. Mathematical description of the responses of simple cortical cells. *Journal of Optical Society of America*, 70:1297–1300, 1980.

[126] E. Marchand. ViSP: A Software Environment for Eye-in-Hand Visual Servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'99*, volume 4, pp. 3224–3229, Detroit, Michigan, Mai 1999.

[127] E. Marchand. Reconstruction d'objets complexes et exploration de scène 3D. Rapport de convention, contrat CEMAGREF-OFIVAL - INRIA Rennes 1.02.C.494,, septembre 2002.

[128] E. Marchand, P. Bouthemy, F. Chaumette. A 2D-3D model-based approach to real-time visual tracking. *Image and Vision Computing, IVC*, 19(13):941–955, novembre 2001.

[129] E. Marchand, F. Chaumette. Active sensor placement for complete scene reconstruction and exploration. In *IEEE Int. Conf. on Robotics and Automation*, volume 1, pp. 743–751, Albuquerque, New Mexico, avril 1997.

[130] E. Marchand, F. Chaumette. An autonomous active vision system for complete and accurate 3D scene reconstruction. Rapport de recherche 1156, IRISA, janvier 1998.

[131] E. Marchand, F. Chaumette. Active vision for complete scene reconstruction and exploration. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(1):65–72, Janvier 1999.

[132] E. Marchand, F. Chaumette. An Autonomous Active Vision System for Complete and Accurate 3D Scene Reconstruction. *Int. Journal of Computer Vision*, 32(3):171–194, août 1999.

[133] E. Marchand, F. Chaumette. A new formulation for non-linear camera calibration using virtual visual servoing. Rapport de recherche 4096, INRIA, janvier 2001.

[134] E. Marchand, F. Chaumette. Virtual Visual Servoing: a framework for real-time augmented reality. In *EUROGRAPHICS'02 Conference Proceeding*, volume 21(3) of *Computer Graphics Forum*, pp. 289–298, Saarebrücken, Germany, septembre 2002.

[135] E. Marchand, F. Chaumette, A. Rizzo. Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'96*, volume 3, pp. 1083–1090, Osaka, Japan, novembre 1996.

[136] E. Marchand, F. Chaumette, F. Spindler, M. Perrier. Controlling an uninstrumented manipulator by visual servoing. *The Int. Journal of Robotics Research, IJRR*, 21(7):635–648, juillet 2002.

[137] E. Marchand, A. Comport, F. Chaumette. Improvements in robust 2D visual servoing. In *IEEE Int. Conf. on Robotics and Automation, ICRA'04*, volume 1, pp. 745–750, New Orleans, avril 2004.

[138] E. Marchand, N. Courty. Controlling a camera in a virtual environment: Visual servoing in computer animation. *The Visual Computer Journal*, 18(1):1–19, février 2002.

[139] E. Marchand, G.-D. Hager. Dynamic Sensor Planning in Visual Servoing. In *IEEE Int. Conf. on Robotics and Automation*, volume 3, pp. 1988–1993, Lueven, Belgium, mai 1998.

[140] E. Marchand, E. Rutten, H. Marchand, F. Chaumette. Specifying and verifying active vision-based robotic systems with the Signal environment. *Int. Journal of Robotics Research*, 17(4):418–432, avril 1998.

[141] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman, San Francisco, 1982.

[142] F. Martin, R. Horaud. Multiple Camera Tracking of Rigid Objects. *Int. Journal of Robotics Research*, 21(2):97–113, février 2002. (Rapport INRIA RR-4268, septembre 2001).

[143] P. Martinet, J. Gallice, D. Khadraoui. Robot control using 3D visual features. In *Word Automation Congress, WAC'96*, volume 3, pp. 497–502, Montpellier, mai 1996.

[144] W. Matusik, C. Buehler, R. Raskar, L. McMillan, S. Gortler. Image-Based Visual Hulls. In *Proceedings of SIGGRAPH 2000*, août 2000.

[145] J. Maver, R. Bajcsy. Occlusions as a guide for planning the next view. *IEEE Trans. on Pattern Analysis and Machine intelligence*, 15(5):417–433, mai 1993.

[146] S.-J. Maybank, O. Faugeras. A theory of self calibration of a moving camera. *Int. Journal of Computer Vision, IJCV*, 8(1):123–152, 1992.

[147] S. Menardais. *Modèle adaptatif multi-niveaux de mouvements humains à partir d'acquisitions*. Thèse de doctorat, Université de Rennes 1, IRISA, 2003.

[148] Y. Mezouar, F. Chaumette. Path Planning For Robust Image-based Control. *IEEE Trans. on Robotics and Automation*, 18(4):534–549, août 2002.

[149] Y. Mezouar, H. Hadj Abdelkader, P. Martinet, F. Chaumette. Visual Servoing from 3D Straight Lines with Central Catadioptric Cameras. In *Fifth Workshop on Omnidirectional Vision, Omnivis'2004*, Prague, Czech Republic, May 2004.

[150] G. Millerson. *Méthode d'éclairage pour le film et la TV*. Édition Dujarric, Paris, 1989.

[151] P. Moutarlier, R. Chatila. Incremental Free-Space Modelling from Uncertain Data by an Autonomous Mobile Robot. In *Workshop on Geometric Reasoning for Perception and Action*, LNCS No 708, Grenoble, France, septembre 1991.

[152] C. Nastar, N. Ayache. Fast segmentation, tracking and analysis of deformable objects. In *Int. Conf. on Computer Vision, ICCV'93*, pp. 275–279, Berlin, Allemagne, 1993.

[153] N. Navab. Industrial Augmented Reality: Challenges in Design and Commercialization of Killer Apps. In *IEEE Int. Symp. on Mixed and Augmented Reality, ISMAR'03*, pp. 2–7, Tokyo, Japan, octobre 2003.

[154] B. Nelson, P.K. Khosla. Integrating sensor placement and visual tracking strategies. In *IEEE Int. Conf. Robotics and Automation*, volume 2, pp. 1351–1356, San Diego, mai 1994.

[155] B. Nelson, P.K. Khosla. The Resolvability Ellipsoid for Visual Servoing. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, pp. 829–832, Seattle, Washington, juin 1994.

[156] B. Nelson, P.K. Khosla. Strategies for Increasing the Tracking Region of an Eye-in-Hand System by Singularity and Joint Limits Avoidance. *Int. Journal of Robotics Research*, 14(3):255–269, juin 1995.

[157] B.J. Nelson, N.P. Papanikolopoulos, P.K. Khosla. Robotics visual servoing and robotic assembly tasks. *IEEE Robotics and Automation Magazine*, 3(2):23–31, juin 1996.

[158] D. Oberkampf, D.F. Dementhon, L.S. Davis. Iterative Pose Estimation Using Coplanar Feature Points. *Computer Vision and Image Understanding*, 63(3):495–511, mai 1996.

[159] J.-M. Odobez, P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, décembre 1995.

[160] J.-M. Odobez, P. Bouthemy, E. Fleuet. Suivi 2D de pièces métalliques en vue d'un asservissement visuel. In *11ème congrès RFIA'98*, volume 2, pp. 173–182, Clermont-Ferrand, Janvier 1998.

[161] N. P. Papanikolopoulos, P. K Khosla. Selection of Features and Evaluation of Visual Measurements for 3D Robotic Visual Tracking. *Int. Symp. on Intelligent Control.*, pp. 320–325, août 1993.

[162] J. Pearl. *Probabilistic reasoning in intelligent systems : Networks of plausible inference*. Morgan Kaufmann Publisher Inc., San Mateo, California, 1988.

[163] T.-Q. Phong, R. Horaud, A. Yassine, P.-D. Tao. Object Pose from a 2D to 3D point and line correspondance. *Int. Journal of Computer Vision*, 15(3):225–243, juillet 1995.

[164] P. Popovič, A. Witkin. Physically Based Motion Transformation. In Alyn Rockwood, Ed., *SIGGRAPH 99, in Computer Graphics Proceedings*, pp. 11–20, ACM Press, New-york, USA, août 1999.

[165] M. Pressigout, E. Marchand. Model-free augmented reality by virtual visual servoing. In *IAPR Int. Conf. on Pattern Recognition, ICPR'04*, Cambridge, UK, août 2004.

[166] R.D. Rimey, C. Brown. Control of Selective Perception Using Bayes Nets and Decision Theory. *Int. Journal of Computer Vision*, 12(2/3):173–207, avril 1994.

[167] D. Robinson. The mechanics of human saccadic eye movements. *Journal of Physiology*, 174:245–264, 1964.

[168] C.F. Rose, B. Guenter, B. Bodenheimer, M.F. Cohen. Efficient Generation of Motion Transitions using Spacetime Constraints. In *SIGGRAPH 96, in Computer Graphics Proceedings*, pp. 147–154, New Orleans, Louisiane, août 1996.

[169] P.J. Rousseeuw, A.M. Leroy. *Robust Regression and Outlier Detection*. John Wiley and Sons, New York, 1987.

[170] A. Ruf, R. Horaud. Rigid and Articulated Motion Seen with an Uncalibrated Stereo Rig. In *IEEE Int. Conf. on Computer Vision*, pp. 789–796, Corfu, Greece, septembre 1999.

[171] E. Rutten, E. Marchand, F. Chaumette. An Experiment with Reactive Data-flow Tasking in Active Robot Vision. *Software - Practices & Experience.*, 27(5):599–621, mai 1997.

[172] R. Safaee-Rad, I. Tchoukanov, B. Benhabib, K.C. Smith. 3D-pose estimation from a quadratic-curved feature in two perspective views. In *IAPR Int. Conf. on Pattern Recognition, ICPR'92*, volume 1, pp. 341–344, La Haye, Pays Bas, août 1992.

[173] C. Samson, M. Le Borgne, B. Espiau. *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991.

[174] J. Santos-Victor, G. Sandini. Visual Behaviors for Docking. *Computer Vision and Image Understanding*, 67(3):223–238, septembre 1997.

[175] S. Seitz, C. Dyer. Photorealistic scene reconstruction by voxel coloring. *Int. J. of Computer Vision*, 35:151–173, février 1999.

[176] R. Sharma, H. Sutanto. A framework for robot motion planning with sensor constraints. *IEEE Trans. on Robotics and Automation*, 13(1):61–73, février 1997.

[177] J. Shi, C. Tomasi. Good Features to Track. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, pp. 593–600, Seattle, Washington, juin 1994.

[178] G. Simon, M.-O. Berger. Registration with a Zoom Lens Camera for Augmented Reality Applications. In *ACM/IEEE Int. Workshop on Augmented Reality, IWAR'99*, pp. 103–114, San Francisco, CA, octobre 1999.

[179] G. Simon, M.-O. Berger. Pose Estimation for Planar Structures. *IEEE Computer Graphics and Applications*, 22(6):46–53, novembre 2002.

[180] G. Simon, A. Fitzgibbon, A. Zisserman. Markerless Tracking using Planar Structures in the Scene. In *IEEE/ACM Int. Symp. on Augmented Reality*, pp. 120–128, Munich, Germany, octobre 2002.

[181] C.-V. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, septembre 1999.

[182] I.H. Suh, T.W. Kim. Visual Servoing of Robot Manipulators by Fuzzy Membership Function Based Neural Networks. In K. Hashimoto, Ed., *Visual Servoing*, volume 7, pp. 285–315. World Scientific Series in Robotics and Automated Systems, Singapour, 1993.

[183] V. Sundareswaran, R. Behringer. Visual Servoing-based Augmented Reality. In *IEEE Int. Workshop on Augmented Reality*, San Francisco, novembre 1998.

[184] H. Sutanto, R. Sharma, V. Varma. The role of exploratory movement in visual servoing without calibration. *Robotics and autonomous system*, 12:153–169, 1998.

[185] M.J. Swain, M.A. Stricker. Promising Direction in Active Vision. *Int. Journal of Computer Vision*, 11(2):109–127, octobre 1993.

[186] S. Tak, O. Song, H. Ko. Motion Balance Filtering. *Computer Graphics Forum*, 19(3):45–61, août 2000.

[187] K. Tarabanis, P.K. Allen, R. Tsai. A Survey of Sensor Planning in Computer Vision. *IEEE Trans. on Robotics and Automation*, 11(1):86–104, février 1995.

[188] K. Tarabanis, R. Tsai, P.K. Allen. The MVP Sensor Planning System for Robotic Vision Tasks. *IEEE Trans. on Robotics and Automation*, 11(1):72–85, février 1995.

[189] J.-P. Tarel, A. Gagalowicz. Calibration de Caméra à Base d'Ellipses. *Traitement du Signal*, 12(2):177–187, 1995.

[190] C.-J. Taylor, J.-P. Ostrowski, S.-H. Jung. Robust Visual Servoing based on Relative Orientation. *Int Conf on Computer Vision and Pattern Recognition*, pp. 574–580, juin 1999.

[191] N. Thalmann, D. Thalmann. *Computer Animation*. Springer Verlag, New York, 1990.

[192] N. Tolani, N. Badler. Real-time inverse kinematics for the human arm. *Presence*, 5(4):393–401, Fall 1996.

[193] T. Tommasini, A. Fusiello, E. Trucco, V. Roberto. Making Good Features Track Better. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp. 178–183, Santa Barbara, USA, juin 1998.

[194] M. Tonko, H.H. Nagel. Model-Based Stereo-Tracking of Non-Polyhedral Objects for Automatic Disassembly Experiments. *Int. Journal of Computer Vision*, 37(1):99–118, juin 2000.

[195] B. Triggs, C. Laugier. Automatic camera placement for robot vision. In *IEEE Int. Conf. on Robotics and Automation*, volume 2, pp. 1732–1738, Nagoya, Japon, mai 1995.

[196] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, août 1987.

[197] X. Tu, D. Terzopoulos. Artificial Fishes: Physics, Locomotion, Perception, Behavior. In *Proceedings of the annual conference on Computer graphics and interactive techniques, ACM Siggraph'94*, pp. 43–50, Orlando, Floride, juillet 1994.

[198] L. Vacchetti, V. Lepetit, P. Fua. Stable 3–d tracking in real-time using integrated context information. In *IEEE Int. Conf. on Conference on Computer Vision and Pattern Recognition, CVPR'03*, volume 2, pp. 241–248, Madison, WI, juin 2003.

[199] M. Vincze. Robust Tracking of Ellipses at Frame Rate. *Pattern Recognition*, 34(2):487 – 498, février 2001.

[200] C.W. Wampler. Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least Squared Method. *IEEE Trans. on Systems, Man, and Cybernetics*, 16(1):93–101, janvier 1986.

[201] C. Ware, S. Osborn. Exploration and virtual camera control in virtual three dimensional environments. In *Symposium on Interactive 3D Graphics*, pp. 175–183, mars 1990.

[202] G.-Q. Wei, S.D. Ma. Implicit and Explicit Camera Calibration: Theory and Experiments. *IEEE Trans. on Pattern Analysis and Machine intelligence*, 16(5):469–480, mai 1994.

[203] J. Weng, P. Cohen, M. Herniou. Camera Calibration with Distorsion Models and Accuracy Evaluation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(10):965–980, octobre 1992.

[204] P. Whaite, F. Ferrie. Autonomous exploration: Driven by uncertainty. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, pp. 339–346, Seattle, Washington, juin 1994.

[205] W. Wilson, C. Hulls, G. Bell. Relative end-effector control using cartesian position-based visual servoing. *IEEE Trans. on Robotics and Automation*, 12(5):684–696, octobre 1996.

[206] A. Witkin, Z. Popović. Motion Warping. In *SIGGRAPH 95, in Computer Graphics Proceedings*, pp. 105–108, Los Angeles, Californie, USA, août 1995.

[207] L.E. Wixson. Viewpoint Selection for Visual Search. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, pp. 800–805, Seattle, Washington, juin 1994.

[208] Y. Ye, J.K. Tsotsos. Sensor Planning for Object Search. *Computer Vision and Image Understanding*, 73(2):145–168, 1999.

[209] V. Zordan, J. Hodgins. Tracking and modifying human motion with dynamic simulation. In *SIGGRAPH 99, in Computer Graphics Proceedings*, pp. 280–280, New-York, USA, août 1999.